JUNOS® Software

MX-series Layer 2 Configuration Guide

Release 9.4

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-280708-01, Revision 1

JUNOS® Software MX-series Layer 2 Configuration Guide
Release 9.4

Writing: Walter Goralski
Editing: Sonia Saruba
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
15 January 2009—Revision 1

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

Part 6 Indexes

List of Tables

About This Guide

This preface provides the following guidelines for using the *JUNOS® Software MX-series Layer 2 Configuration Guide*:

Objectives

This guide is designed for network administrators who are configuring and monitoring the Layer 2 services supported on a Juniper Networks MX-series router.

NOTE: This guide documents Release 9.4 of the JUNOS software. For additional information about the JUNOS software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks M-series, MX-series, T-series, EX-series, or J-series router or switch.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)

- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Supported Routing Platforms

For the Layer 2 features described in this manual, the JUNOS software currently supports the following routing platforms:

- MX-series

Using the Indexes

This reference contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines,* refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host> show chassis alarms` `No alarms currently active`

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

Convention	Description	Examples
Italic text like this	■ Introduces important new terms. ■ Identifies book names. ■ Identifies RFC and Internet draft titles.	■ A policy *term* is a named structure that defines match conditions and actions. ■ *JUNOS System Basics Configuration Guide* ■ RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Plain text like this	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	■ To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. ■ The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric *metric*>;
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast | multicast (*string1* | *string2* | *string3*)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [*community-ids*]

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	■ In the Logical Interfaces box, select **All Interfaces**. ■ To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols > Ospf**.

List of Technical Publications

Table 3 on page xxii lists the software and hardware guides and release notes for Juniper Networks M-series, MX-series, and T-series routing platforms and describes the contents of each document. Table 4 on page xxvi lists the books included in the *Network Operations Guide* series. Table 5 on page xxvii lists the manuals and release notes supporting JUNOS software for J-series and SRX-series platforms. All documents are available at http://www.juniper.net/techpubs/.

Table 6 on page xxviii lists additional books on Juniper Networks solutions that you can order through your bookstore. A complete list of such books is available at http://www.juniper.net/books.

Table 3: Technical Documentation for Supported Routing Platforms

Book	Description
JUNOS Software for Supported Routing Platforms	
Access Privilege	Explains how to configure access privileges in user classes by using permission flags and regular expressions. Lists the permission flags along with their associated command-line interface (CLI) operational mode commands and configuration statements.
Broadband Subscriber Management Solutions	Describes residential subscriber management and how you can deploy solutions that include multisubscriber IP address assignment, service provisioning, authentication, authorization, accounting, and dynamic request services in your network
Class of Service	Provides an overview of the class-of-service (CoS) functions of the JUNOS software and describes how to configure CoS features, including configuring multiple forwarding classes for transmitting packets, defining which packets are placed into each output queue, scheduling the transmission service level for each queue, and managing congestion through the random early detection (RED) algorithm.
CLI User Guide	Describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Feature Guide	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
High Availability	Provides an overview of hardware and software resources that ensure a high level of continuous routing platform operation and describes how to configure high availability (HA) features such as nonstop active routing (NSR) and graceful Routing Engine switchover (GRES).
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast Protocols	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
Multiplay Solutions	Describes how you can deploy IPTV and voice over IP (VoIP) services in your network.
MX-series Layer 2 Configuration Guide	Provides an overview of the Layer 2 functions of the MX-series routers, including configuring bridging domains, MAC address and VLAN learning and forwarding, and spanning-tree protocols. It also details the routing instance types used by Layer 2 applications. All of this material was formerly covered in the *JUNOS Routing Protocols Configuration Guide*.
MX-series Layer 2 Solutions Guide	Describes common configuration scenarios for the Layer 2 features supported on the MX-series routers, including basic bridged VLANs with normalized VLAN tags, aggregated Ethernet links, bridge domains, Multiple Spanning Tree Protocol (MSTP), and integrated routing and bridging (IRB).
Network Interfaces	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, and forwarding options.
Protected System Domain	Provides an overview of the JCS 1200 platform and the concept of Protected System Domains (PSDs). The JCS 1200 platform, which contains up to six redundant pairs of Routing Engines running JUNOS software, is connected to a T320 router or to a T640 or T1600 routing node. To configure a PSD, you assign any number of Flexible PIC concentrators (FPCs) in the T-series routing platform to a pair of Routing Engines on the JCS 1200 platform. Each PSD has the same capabilities and functionality as a physical router, with its own control plane, forwarding plane, and administration.
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing instances, and unicast routing protocols.
Secure Configuration Guide for Common Criteria and JUNOS-FIPS	Provides an overview of secure Common Criteria and JUNOS-FIPS protocols for the JUNOS software and describes how to install and configure secure Common Criteria and JUNOS-FIPS on a routing platform.
Services Interfaces	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the router.
Software Installation and Upgrade Guide	Describes the JUNOS software components and packaging and explains how to initially configure, reinstall, and upgrade the JUNOS system software. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
Subscriber Access	Provides an overview of the subscriber access features of the JUNOS software and describes how to configure subscriber access support on the router, including dynamic profiles, class of service, AAA, and access methods.
System Basics	Describes Juniper Networks routing platforms and explains how to configure basic system parameters, supported protocols and software processes, authentication, and a variety of utilities for managing your router on the network.
VPNs	Provides an overview and describes how to configure Layer 2 and Layer 3 virtual private networks (VPNs), virtual private LAN service (VPLS), and Layer 2 circuits. Provides configuration examples.
JUNOS References	
Hierarchy and RFC Reference	Describes the JUNOS configuration mode commands. Provides a hierarchy reference that displays each level of a configuration hierarchy, and includes all possible configuration statements that can be used at that level. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Interfaces Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
Routing Protocols and Policies Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing policies and protocols, including firewall filters.
System Basics and Services Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot system basics, including commands for real-time monitoring and route (or path) tracing, system software management, and chassis management. Also describes commands for monitoring and troubleshooting services such as class of service (CoS), IP Security (IPsec), stateful firewalls, flow collection, and flow monitoring.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
J-Web User Guide	
J-Web Interface User Guide	Describes how to use the J-Web graphical user interface (GUI) to configure, monitor, and manage Juniper Networks routing platforms.
JUNOS API and Scripting Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
JUNOS XML API Configuration Reference	Provides reference pages for the configuration tag elements in the JUNOS XML API.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
JUNOS XML API Operational Reference	Provides reference pages for the operational tag elements in the JUNOS XML API.
NETCONF API Guide	Describes how to use the NETCONF API to monitor and configure Juniper Networks routing platforms.
JUNOS Configuration and Diagnostic Automation Guide	Describes how to use the commit script and self-diagnosis features of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts, how to use commit script macros to provide simplified aliases for frequently used configuration statements, and how to configure diagnostic event policies.
Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
PIC Guide	Describes the routing platform's Physical Interface Cards (PICs). Each platform has its own PIC guide.
DPC Guide	Describes the Dense Port Concentrators (DPCs) for all MX-series routers.
JUNOScope Documentation	
JUNOScope Software User Guide	Describes the JUNOScope software graphical user interface (GUI), how to install and administer the software, and how to use the software to manage routing platform configuration files and monitor routing platform operations.
Advanced Insight Solutions (AIS) Documentation	
Advanced Insight Solutions Guide	Describes the Advanced Insight Manager (AIM) application, which provides a gateway between JUNOS devices and Juniper Support Systems (JSS) for case management and intelligence updates. Explains how to run AI-Scripts on Juniper Networks devices.
Release Notes	
JUNOS Release Notes	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS, JUNOScript, and NETCONF manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
Hardware Release Notes	Describe the available documentation for the routing platform and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
JUNOScope Release Notes	Contain corrections and updates to the published JUNOScope manual, provide information that might have been omitted from the manual, and describe upgrade and downgrade procedures.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
AIS Release Notes	Summarize AIS new features and guidelines, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide initial setup, upgrade, and downgrade procedures.
AIS AI-Scripts Release Notes	Summarize AI-Scripts new features, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide instructions for automatic and manual installation, including deleting and rolling back.

Table 4: JUNOS Software Network Operations Guides

Book	Description
Baseline	Describes the most basic tasks for running a network using Juniper Networks products. Tasks include upgrading and reinstalling JUNOS software, gathering basic system management information, verifying your network topology, and searching log messages.
Interfaces	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
MPLS	Describes tasks for configuring, monitoring, and troubleshooting an example MPLS network. Tasks include verifying the correct configuration of the MPLS and RSVP protocols, displaying the status and statistics of MPLS running on all routing platforms in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.
MPLS Log Reference	Describes MPLS status and error messages that appear in the output of the show mpls lsp extensive command. The guide also describes how and when to configure Constrained Shortest Path First (CSPF) and RSVP trace options, and how to examine a CSPF or RSVP failure in a sample network.
MPLS Fast Reroute	Describes operational information helpful in monitoring and troubleshooting an MPLS network configured with fast reroute (FRR) and load balancing.
Hardware	Describes tasks for monitoring M-series and T-series routing platforms.

To configure and operate a J-series Services Router or an SRX-series Services Gateway running JUNOS software, you must also use the configuration statements and operational mode commands documented in JUNOS configuration guides and command references. To configure and operate a WX Integrated Services Module, you must also use WX documentation.

Table 5: JUNOS Software for J-series Services Routers and SRX-series Services Gateways Documentation

Book	Description
J-series and SRX-series Platforms	
JUNOS Software Interfaces and Routing Configuration Guide	Explains how to configure SRX-series and J-series interfaces for basic IP routing with standard routing protocols, ISDN service, firewall filters (access control lists), and class-of-service (CoS) traffic classification.
JUNOS Software Security Configuration Guide	Explains how to configure and manage SRX-series and J-series security services such as stateful firewall policies, IPsec VPNs, firewall screens, Network Address Translation (NAT), Public Key Cryptography, chassis clusters, Application Layer Gateways (ALGs), and Intrusion Detection and Prevention (IDP).
JUNOS Software Administration Guide	Shows how to monitor SRX-series and J-series devices and routing operations, firewall and security services, system alarms and events, and network performance. This guide also shows how to administer user authentication and access, upgrade software, and diagnose common problems.
JUNOS Software CLI Reference	Provides the complete configuration hierarchy available on SRX-series and J-series devices. This guide also describes the configuration statements and operational mode commands unique to these devices.
JUNOS Release Notes	Summarize new features and known problems for a particular release of JUNOS software, including JUNOS software for J-series and SRX-series devices.The release notes also contain corrections and updates to the manuals and software upgrade and downgrade instructions for JUNOS software.
J-series Only	
JUNOS Software Design and Implementation Guide	Provides guidelines and examples for designing and implementing IPsec VPNs, firewalls, and routing on J-series Services Routers running JUNOS software.
J-series Services Routers Quick Start	Explains how to quickly set up a J-series Services Router. This document contains router declarations of conformity.
JUNOS Software with Enhanced Services J-series Services Router Hardware Guide	Provides an overview, basic instructions, and specifications for J-series Services Routers. This guide explains how to prepare a site, unpack and install the router, replace router hardware, and establish basic router connectivity. This guide contains hardware descriptions and specifications.
JUNOS Software Migration Guide	Provides instructions for migrating an SSG device running ScreenOS software to JUNOS software or upgrading a J-series device to a later version of the JUNOS software.
WXC Integrated Services Module Installation and Configuration Guide	Explains how to install and initially configure a WXC Integrated Services Module in a J-series Services Router for application acceleration.

Table 6: Additional Books Available Through http://www.juniper.net/books

Book	Description
Interdomain Multicast Routing	Provides background and in-depth analysis of multicast routing using Protocol Independent Multicast sparse mode (PIM SM) and Multicast Source Discovery Protocol (MSDP); details any-source and source-specific multicast delivery models; explores multiprotocol BGP (MBGP) and multicast IS-IS; explains Internet Gateway Management Protocol (IGMP) versions 1, 2, and 3; lists packet formats for IGMP, PIM, and MSDP; and provides a complete glossary of multicast terms.
JUNOS Cookbook	Provides detailed examples of common JUNOS software configuration tasks, such as basic router configuration and file management, security and access control, logging, routing policy, firewalls, routing protocols, MPLS, and VPNs.
MPLS-Enabled Applications	Provides an overview of Multiprotocol Label Switching (MPLS) applications (such as Layer 3 virtual private networks [VPNs], Layer 2 VPNs, virtual private LAN service [VPLS], and pseudowires), explains how to apply MPLS, examines the scaling requirements of equipment at different points in the network, and covers the following topics: point-to-multipoint label switched paths (LSPs), DiffServ-aware traffic engineering, class of service, interdomain traffic engineering, path computation, route target filtering, multicast support for Layer 3 VPNs, and management and troubleshooting of MPLS networks.
OSPF and IS-IS: Choosing an IGP for Large-Scale Networks	Explores the full range of characteristics and capabilities for the two major link-state routing protocols: Open Shortest Path First (OSPF) and IS-IS. Explains architecture, packet types, and addressing; demonstrates how to improve scalability; shows how to design large-scale networks for maximum security and reliability; details protocol extensions for MPLS-based traffic engineering, IPv6, and multitopology routing; and covers troubleshooting for OSPF and IS-IS networks.
Routing Policy and Protocols for Multivendor IP Networks	Provides a brief history of the Internet, explains IP addressing and routing (Routing Information Protocol [RIP], OSPF, IS-IS, and Border Gateway Protocol [BGP]), explores ISP peering and routing policies, and displays configurations for both Juniper Networks and other vendors' routers.
The Complete IS-IS Protocol	Provides the insight and practical solutions necessary to understand the IS-IS protocol and how it works by using a multivendor, real-world approach.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version (not required for *Network Operations Guides [NOGs])*

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 JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/customers/support/downloads/710059.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

Part 1
Overview

Chapter 1

Overview of Layer 2 Services for MX-series Routers

This chapter provides the following information about configuring Layer 2 services for MX-series routers:

MX-series Architecture

The key components of each MX-series router are Dense Port Concentrators (DPCs), the Routing Engine, and the Switch Control Board.

The DPCs are optimized for Ethernet density and are capable of supporting up to 40 Gigabit Ethernet or 4 10-Gigabit Ethernet ports. The DPC assembly combines packet forwarding and Ethernet interfaces on a single board, with four 10-Gbps Packet Forwarding Engines. Each Packet Forwarding Engine consists of one chip for Layer 3 processing and one Layer 2 network processor. The DPCs interface with the power supplies and Switch Control Boards (SCBs).

The Routing Engine is an Intel-based PC platform that runs the JUNOS Operating System. Software processes that run on the Routing Engine maintain the routing tables, manage the routing protocols used on the router, control the router interfaces, control some chassis components, and provide the interface for system management and user access to the router. Routing Engines communicate with DPCs via dedicated out-of-band management channels, providing a clear distinction between the controls and forwarding planes.

The Switch Control Board (SCB) powers cards on and off; controls clocking, resets and booting; and monitors and controls systems functions, including fan speed, board power status, PDM status and control, and the system front panel. Integrated into the SCB is the switch fabric, which interconnects all of the DPCs within the chassis, supporting up to 48 Packet Forwarding Engines. The Routing Engine installs directly into the SCB.

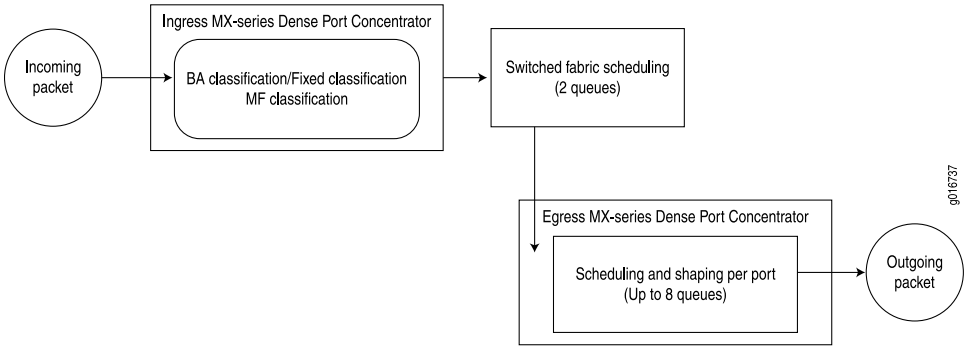
The MX-series router has been optimized for Ethernet services. Examples of the wide range of Ethernet services provided by the MX-series include:

- Virtual private LAN service (VPLS) for multipoint connectivity—Native support for VPLS services
- Virtual leased line (VLL) for point-to-point services—Native support for point-to-point services
- RFC 2547.bis IP/MPLS VPN (L3VPN)—Full support for MPLS VPNs throughout the Ethernet network
- Video distribution IPTV services
- Ethernet aggregation at the campus/enterprise edge—Supports dense 1–Gigabit Ethernet and 10–Gigabit Ethernet configurations, and provides full Layer 3 support for campus edge requirements
- Ethernet aggregation at the multiservice edge—Supports up to 480 1–Gigabit Ethernet ports or 48 10–Gigabit Ethernet ports for maximum Ethernet density along, with full Layer 2 and Layer 3 VPN support for MSE applications

Architecture Features

The architecture for MX-series routing platforms such as the MX960 Ethernet Services Router is similar in concept, but different in particulars, from other routing platforms. The general architecture for the MX-series routing platform is shown in Figure 1 on page 4.

Figure 1: MX-series Packet Forwarding and Data Flow



DPCs

MX-series routers process incoming and outgoing packets with the DPC. The MX-series routers do not use the more traditional Flexible Port Concentrators (FPCs) that are used by the T-series routing platforms and many of the M-series routing platforms. FPCs are populated with physical interface cards (PICs) for various interface types. The DPC supports up to one 40-Gigabit Ethernet or four 10-Gigabit Ethernet ports, combining these ports with four 10-Gbps *Packet Forwarding Engines* on a single interface card, combining the functions of four FPCs and the PICs.

The MX960 has 12 DPC slots. The MX480 has 7 DPC slots. The MX240 has 4 DPC slots. Each DPC has either 40 Gigabit Ethernet ports or 4 10-Gigabit Ethernet ports.

NOTE: MX-series routers use DPCs rather than FPCs and therefore do not support Physical Interface Cards (PICs). In the JUNOS CLI, however, you use the FPC syntax to configure or display information about DPCs, and you use the PIC syntax to configure or display information about Packet Forwarding Engines on the DPCs.

In addition to Layer 3 routing capabilities, the DPCs also have many Layer 2 functions that allow MX-series routing platforms to be used for many virtual LAN (VLAN) and other Layer 2 network applications.

MX-series Layer 3 and Layer 2 Functions and Configuration

You can configure Layer 2 or Layer 3 features and functions on MX-series routers. This book discusses Layer 2 configurations, including Layer 2 statement summaries and configuration statement examples. For more complete configuration examples, see the *MX-series Solutions Guide*.

For more information about configuring Layer 3 features and functions (such as class of service), see the relevant JUNOS configuration guides.

MX-series Snooping

MX-series routers can support both Layer 3 and Layer 2 functions at the same time. For example, you can configure the Layer 3 multicast protocols Protocol Independent Multicast (PIM) and the Internet Group Membership Protocol (IGMP) as well as Layer 2 VLANs on the MX-series router. In many cases, Layer 2 protocols run on some interfaces, and Layer 2 protocols run on others.

Normal encapsulation rules restrict Layer 2 processing to accessing information in the frame header and Layer 3 processing to accessing information in the packet header. However, in some cases, an interface running a Layer 2 protocol needs information available only at Layer 3. For example, in multicast applications, the VLANs need the group membership information and multicast tree information available to the Layer 3 IGMP and PIM protocols. In these cases, the Layer 3 configurations can use PIM or IGMP snooping to provide the needed information at the VLAN level.

Snooping configuration statements and examples are not included in this configuration guide. For more information about configuring PIM and IGMP snooping, see the *JUNOS Multicast Configuration Guide*.

Part 2
Configuration Basics for Layer 2 Services

Chapter 2

Configuring Routing Instances for MX-series Layer 2 Services

This chapter describes the routing instance types used by Layer 2 services on MX-series routers.

Routing Instances Overview

A routing instance is a routing entity for a router. You can create multiple instances of BGP, IS-IS, OSPF, OSPFv3, RIP, and static routes. Each instance contains a routing table, applied routing policies, routing table group, interfaces that belong to that instance, and a protocol-specific route configuration related to that instance.

You configure a primary routing instance at the [edit protocols] hierarchy level. You configure additional routing instances at the [edit routing-instances] or [edit logical-systems *logical-system-name* routing-instance] hierarchy level.

You use routing instances to:

- Create administrative separation in a large network to segregate customer traffic and associated settings. The customers see only the routes belonging to them.
- Create overlay networks in which separate services are routed only towards routers participating in that service, such as voice. The overlay network isolates routes belonging to one service from another service by exporting routes, applying tags, and filtering based on tags.

Each routing instance consists of sets of the following:

- A set of routing tables
- A set of interfaces that belong to these routing tables
- A set of routing option configurations

Each routing instance has a unique name and a corresponding IP unicast table. For example, if you configure a routing instance with the name my-instance, its

corresponding IP unicast table will be my-instance.inet.0. All routes for my-instance are installed into my-instance.inet.0.

Routes are installed into the default routing instance inet.0 by default, unless a routing instance is specified.

For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Routing Instances Basic Configuration

To configure routing instances, include the following statements:

```
routing-instances {
   routing-instance-name {
      description text;
      forwarding-options;
      interface interface-name;
      instance-type (forwarding | layer2-control | l2vpn | no-forwarding | virtual-router |
         virtual-switch | vpls | vrf);
      bridge-domains {
         bridge-domains-name {
            domain-type bridge;
            vlan-id (none | all | number);
            vlan-tags outer number inner number;
            interface interface-name;
            routing-interface routing-interface-name;
            bridge-options {
               interface-mac-limit limit;
               mac-statistics;
               mac-table-size limit;
               no-mac-learning;
               static-mac mac-address;
            }
         }
      }
      no-vrf-advertise;
      route-distinguisher (as-number:number | ip-address:number);
      vrf-import [ policy-names ];
      vrf-export [ policy–names ];
      vrf-table-label;
      vrf-target {
         export community-name;
         import community-name;
      }
      protocols {
         ... protocol-configuration ...
      }
      routing-options {
         ... routing-options ...
      }
   }
}
```

With the exception of the virtual-switch routing instance type (instance-type virtual-switch), you can include the statements at the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

The instance-type virtual-switch statement is not supported at the [edit logical-systems *logical-system-name*] hierarchy level.

You can configure eight types of routing instances:

- Forwarding instance—For more information about the forwarding routing instance type, see the *JUNOS Routing Protocols Configuration Guide*.
- Layer 2 VPN routing instance—For more information about the l2vpn routing instance type, see the *JUNOS VPNs Configuration Guide*.
- Layer 2 control protocols—(MX-series routers only) For more information about the layer2-control routing instance type, see "Configuring Routing Instance Types Used in Layer 2 Networking" on page 11 and "Configuring Spanning-Tree Protocols" on page 87.
- Nonforwarding instance—For more information about the no-forwarding routing instance type, see the *JUNOS Routing Protocols Configuration Guide*.
- Virtual routing instance—For more information about the virtual-router routing instance type, see the *JUNOS Routing Protocols Configuration Guide*.
- Virtual switch routing instance—(MX-series routers only) For more information about the virtual-switch routing instance type, see "Configuring Routing Instance Types Used in Layer 2 Networking" on page 11 and "Configuring Layer 2 Bridging" on page 37.
- Virtual private LAN service (VPLS) routing instance—For more information about the vpls routing instance type, see "Configuring Routing Instance Types Used in Layer 2 Networking" on page 11 and "Configuring Layer 2 Bridging" on page 37.
- VPN routing and forwarding (VRF) instance—For more information about the vrf routing instance type, see the *JUNOS Routing Protocols Configuration Guide*.

NOTE: In a Layer 2 network, you can configure only three types of routing instances: Layer 2 control protocols, VPLS routing instance, and virtual switch routing instance.

Configuring Routing Instance Types Used in Layer 2 Networking

Although routing instances are primarily intended to maintain separation of tables and protocols at Layer 3 (mirroring the traditional IP network separation at Layer 3), many aspects of routing instances make them convenient to use for Layer 2 applications and architectures as well. In Layer 2 applications, routing instances still help to maintain table, interface, and customer insulation, but with regard to media access control (MAC) addresses and VLAN tags as much as IP addresses.

In summary, some routing instance types are most useful for system configurations concerning Layer 3 (IP) networking, and other routing instance types are most useful for configurations concerning Layer 2 (VLAN) networking. This document describes the routing instance types that are used for Layer 2 applications. For more information about other types of routing instances, see the *JUNOS Routing Protocols Configuration Guide*.

Three types of routing instances can be used on an MX-series router for Layer 2 networking:

Layer 2 Control Protocols

On MX-series routers only, use the layer2-control routing instance type for Rapid Spanning Tree Protocol (RSTP) or Multiple Spanning Tree Protocol (MSTP) in customer edge interfaces of a VPLS routing instance. Layer 2 control protocols enable features such as Layer 2 protocol tunneling or nonstop bridging. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer edge interface is multihomed to two provider edge interfaces, use the default bridge protocol data unit (BPDU) tunneling.

To create a routing instance for Layer 2 control protocols, you must include at least the following statements in the configuration:

```
routing-instances {
    routing-instance-name {
        instance-type layer2-control;
        interface interface-name;
        route-distinguisher (as-number:number | ip-address:number);
        vrf-import [ policy-names ];
        vrf-export [ policy-names ];
        protocols {
            mstp {
                ... interface options ...
                msti msti-id {
                    ... MSTP MSTI configuration ...
                }
            }
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

Virtual Switch Routing Instance

On MX-series routers only, use the virtual-switch routing instance type to isolate a LAN segment with its STP 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), RSTP, and MSTP.

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;
                vlan-id (all | none | number);
                vlan-tags outer number inner number;
                interface interface-name;
            }
        }
        protocols {
            mstp ...
        }
    }
}
```

The instance-type virtual-switch statement is not supported at the [edit logical-systems *logical-system-name*] hierarchy level.

For more information about configuring virtual switches, see "Configuring Layer 2 Virtual Switches" on page 45.

VPLS Routing Instance

Use the vpls routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN.

To create a routing instance for VPLS, you must include at least the following statements in the configuration:

```
routing-instances {
    routing-instance-name {
        instance-type vpls;
        interface interface-name;
        route-distinguisher (as-number:number | ip-address:number);
        vrf-import [ policy-names ];
        vrf-export [ policy-names ];
        protocols {
            vpls {
```

```
            ... vpls configuration ...
        }
    }
}
}
```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

For more information about configuring VPLS, see the *JUNOS VPNs Configuration Guide*. For a detailed VPLS example configuration, see the *JUNOS Feature Guide*.

Chapter 3

Configuring Layer 2 Port Mirroring

This chapter describes the Layer 2 port mirroring feature supported on MX-series routers.

Layer 2 Port Mirroring Overview

On routing platforms that contain an Internet Processor II ASIC, you can send a copy of any incoming packet from the routing platform to an external host address or a packet analyzer for analysis. This is known as port mirroring.

Beginning with JUNOS Release 9.3, MX-series routers support port mirroring for Layer 2 bridging traffic. Layer 2 port mirroring enables you to specify the manner in which incoming and outgoing packets at specified ports in a bridging environment are monitored and sampled and the manner in which copies of the sampled packet are forwarded to another destination, where the packets can be analyzed.

MX-series routers support Layer 2 port mirroring by performing flow monitoring functions using a class-of-service (CoS) architecture that is similar in concept, but different in particulars, from other routing platforms. For general information about packet flow within MX-series platforms and other routing platforms, see the *JUNOS Class of Service Configuration Guide*.

In a Layer 2 environment, MX-series routers support port mirroring of VPLS (family bridge or family vpls) traffic. MX-series routers also support port mirroring for Layer 2 VPNs (L2 VPNs) with family ccc. In a Layer 3 environment, MX-series routers support port mirroring of IPv4 (family inet) and IPv6 (family inet6) traffic. Like the M120 and

M320 routers, MX-series routers support port mirroring of IPv4, IPv6, and VPLS packets simultaneously.

This chapter describes port mirroring of Layer 2 bridging traffic that passes through an MX-series router. For information about Layer 3 port mirroring, see the *JUNOS Policy Framework Configuration Guide*.

Layer 2 Port Mirroring Features

This section describes the features of Layer 2 port mirroring:

Different Port-Mirroring Properties for Different Router Interfaces

You can configure different sets of Layer 2 port-mirroring properties, known as port-mirroring instances, for different interfaces on an MX-series router:

- All ports in the chassis—The set of Layer 2 port mirroring properties configured at the [edit forwarding-options port-mirroring] hierarchy level is known as the global port-mirroring instance. If configured, these properties implicitly apply to all VPLS packets received on all ports in the router chassis. For detailed configuration information, see "Layer 2 Port Mirroring for the Global Instance" on page 18.
- Ports for a specific DPC or Packet Forwarding Engine—You can configure multiple, named port-mirroring instances, with each instance specifying different input sampling properties and output mirror destination properties. A named port-mirroring instance can be applied to a specific DPC to override the port-mirroring properties configured by the global port-mirroring instance. A named port-mirroring instance can also be applied to a specific Packet Forwarding Engine to override the port-mirroring properties configured for the DPC or for the global port-mirroring instance. For detailed configuration information, see "Layer 2 Port Mirroring for a DPC or a Packet Forwarding Engine" on page 19.
- A logical interface or a bridge domain forwarding table—You can configure a Layer 2 port-mirroring firewall filter that can be applied to a logical interface (including an aggregated Ethernet interface), the forwarding table of a bridge domain, or the flood table of a VPLS routing instance. A Layer 2 port-mirroring firewall filter uses the input sampling properties and output mirror destination properties configured in the global port-mirroring instance. In a Layer 2 port-mirroring firewall filter configuration, you can include one or more actions (under the then statement along with the port-mirror action modifier) that are to be taken on the mirrored packets. For detailed configuration information, see "Layer 2 Port Mirroring for a Logical Interface, Forwarding Table, or Flood Table" on page 23.

Input Packet-Sampling Properties

The input packet-sampling properties of Layer 2 port-mirroring instance specify how the input packets are to be selected for mirroring:

- The rate specifies the number of packets in each sample.
- The run-length specifies the number of packets to mirror from each sample.

Mirror Destination Properties

The mirror destination properties of a Layer 2 port-mirroring instance specify the destination of the mirrored packets:

- The number of port-mirroring destinations supported for an MX-series router is limited to the number of Packet Forwarding Engines contained on the dense port concentrators (DPCs) installed in the router chassis. To determine the number and type of DPCs in an MX-series router chassis, use the show chassis hardware command.
- If port mirroring is enabled at both ingress and egress interfaces, you can prevent the MX-series router from sending duplicate packets to the same destination (which would complicate the analysis of the mirrored traffic) by enabling the mirror-once option.

NOTE: In typical applications, you send the sampled packets to an analyzer or a workstation for analysis, not to another router. If you must send this traffic over a network, you should use tunnels. For Layer 2 VPN implementations, you can use the Layer 2 VPN routing instance type l2vpn to tunnel the packets to a remote destination. For information about configuring a routing instance for Layer 2 VPN, see the *JUNOS VPNs Configuration Guide*. For a detailed Layer 2 VPN example configuration, see the *JUNOS Feature Guide*. For information about tunnel interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Layer 2 Port-Mirroring Restrictions

The following restrictions apply to Layer 2 port mirroring:

- The port-mirrored input interface should not participate in any kind of routing activity.
- The mirror destination device should be on a dedicated bridge domain and should not participate in any bridging activity. The mirror destination device should not have a route to the ultimate traffic destination, and the mirror destination device should not send the sampled packets back to the source address.

  For example, if the VPLS packets sampled at 190.68.20.5 have a destination address of 190.68.9.10 and the port-mirrored traffic is sent to 190.68.20.15 for analysis, the device associated with 190.68.20.15 should not know a route to 190.68.9.10 and should not send the mirrored packets back to 190.68.20.5.

- Only Layer 2 transit data can be mirrored. Packets generated by the Routing Engine (such as Layer 2 control packets) are not mirrored.
- For either the global port-mirroring instance or a named port-mirroring instance, you can configure only one mirror output interface per port-mirroring instance and packet address family. If you include more than one **interface** statement under the **family (bridge | vpls) output** statement, the previous **interface** statement is overridden.
- Layer 2 port mirroring of input or output to a logical interface, input to a forwarding table in a bridge domain, or input to a flood table for a VPLS routing instance is not supported for logical systems.

Layer 2 Port Mirroring for the Global Instance

This section describes how to configure the set of Layer 2 port-mirroring properties that apply to all ports in the chassis:

- Configuring Layer 2 Port Mirroring for the Global Instance on page 18
- Enabling Mirror-Once Mode for Layer 2 Port Mirroring on page 19

Configuring Layer 2 Port Mirroring for the Global Instance

To configure global port-mirroring properties for a Layer 2 packet address family, include the **input** statement and the **family (bridge | ccc | vpls) output** statement at the **[edit forwarding-options port-mirroring]** hierarchy level:

```
[edit]
forwarding-options {
    port-mirroring {
        input { # Input packet-sampling properties
            maximum-run-length number;
            rate number;
            run-length number;
        }
        family ( bridge | ccc | vpls) {
            output { # Mirror destination properties
                interface interface-name;
                no-filter-check; # Optional
            }
        }
    }
}
```

To configure input packet-sampling properties, include the **input** statement at the **[edit forwarding-options port-mirroring]** hierarchy level. To specify the number of packets in a sample, include the **rate** *number* statement. To specify the number of sampled packets to mirror, include the **run-length** *number* statement.

To configure the length to which mirrored packets are truncated, include the **maximum-packet-length** statement at the **[edit forwarding-options port-mirroring input]** hierarchy level. The default value is **0**, which means the mirrored packets are not truncated. The valid range is **0** through **9216**.

To configure the mirror destination properties, include the family (bridge | ccc | vpls) output statement at the [edit forwarding-options port-mirroring] hierarchy level. To specify the physical interface on which to send the duplicate packets, include the interface *interface-name* statement.

NOTE: Under the [edit forwarding-options port-mirroring] hierarchy level, the protocol family statement family bridge is an alias for family vpls. The command-line interface (CLI) displays Layer 2 port-mirroring configurations as family vpls, even for Layer 2 port-mirroring configured as family bridge.

If you need to allow configuration of filters on the destination interface for the global port-mirroring instance, include the no-filter-check statement. If you apply a filter to an interface that is a Layer 2 port-mirroring destination, a commit failure is returned unless you included the no-filter-check option at the [edit forwarding-options port-mirroring family (bridge | ccc | vpls) output] hierarchy level.

Enabling Mirror-Once Mode for Layer 2 Port Mirroring

When an MX-series router is configured to Layer 2 port mirroring at both ingress and egress interfaces, and the same packet could be mirrored twice. You can configure an MX-series router to mirror traffic only once, so that the router does not send duplicate sampled packets to the same mirroring destination. To configure, include the mirror-once statement at the [edit forwarding-options port-mirroring] hierarchy level.

```
[edit]
forwarding-options {
   port-mirroring {
      mirror-once; # Mirror destinations do not receive duplicate packets
      input {
         ... input-sampling-configuration ...
      }
      family ( bridge | ccc | vpls) {
         output {
            ... mirroring-destination-configuration ...
         }
      }
   }
}
```

Layer 2 Port Mirroring for a DPC or a Packet Forwarding Engine

You can configure multiple instances of Layer 2 port mirroring to enable different Packet Forwarding Engines to mirror packets to different destinations. You can bind a port-mirroring instance to a specific DPC or to a specific Packet Forwarding Engine.

NOTE: MX-series routers use DPCs rather than FPCs and therefore do not support Physical Interface Cards (PICs). In the JUNOS CLI, however, you use the FPC syntax to configure or display information about DPCs, and you use the PIC syntax to configure or display information about Packet Forwarding Engines on the DPCs.

The following sections describe how to configure Layer 2 port mirroring for a specific DPC or Packet Forwarding Engine:

Configuring Layer 2 Port-Mirroring Instances

A Layer 2 port-mirroring instance is a named set of port-mirroring properties that you can associate with a particular Packet Forwarding Engine to mirror packets to different destinations. To configure a Layer 2 port-mirroring instance, include the instance *pm-instance-name* statement at the [edit forwarding-options port-mirroring] hierarchy level:

```
[edit]
forwarding-options {
    port-mirroring {
        instance {
            pm-instance-name-a { # One port-mirroring instance for this router
                input {
                    maximum-run-length number;
                    rate number;
                    run-length number;
                }
                family (bridge | ccc | vpls) {
                    output {
                        interface interface-name;
                        no-filter-check; # Optional
                    }
                }
            }
            .
            .
            .
            pm-instance-name-z { # Another port-mirroring instance for this router
                input {
                    maximum-run-length number;
                    rate number;
                    run-length number;
                }
                family (bridge | ccc | vpls) {
                    output {
                        interface interface-name;
                        no-filter-check; # Optional
                    }
                }
            }
        }
    }
}
```

To configure input packet-sampling properties, include the input statement at the [edit forwarding-options port-mirroring instance *pm-instance-name*] hierarchy level. To specify the number of packets in a sample, include the rate *number* statement. To specify the number of sampled packets to mirror, include the run-length *number* statement.

To configure the mirror destination properties, include the family (bridge | vpls) output statement at the [edit forwarding-options port-mirroring instance *pm-instance-name*] hierarchy level. To specify the physical interface on which to send the duplicate packets, include the interface *interface-name* statement.

NOTE: Under the [edit forwarding-options port-mirroring instance *pm-instance-name*] hierarchy level, the protocol family statement family bridge is an alias for family vpls. The CLI displays Layer 2 port-mirroring configurations as family vpls, even for Layer 2 port-mirroring configured as family bridge.

If you need to allow configuration of filters on the destination interface for a named port-mirroring instance, include the no-filter-check statement. If you apply a filter to an interface that is a Layer 2 port-mirroring destination, a commit failure is returned unless you included the no-filter-check option at the [edit forwarding-options port-mirroring instance *pm-instance-name* family (bridge | vpls) output] hierarchy level.

Determining the Number of DPCs in an MX-series Router

To display information about the number and types of DPCs in an MX-series router, the number of Packet Forwarding Engines on each DPC, and the number and types of ports per Packet Forwarding Engine, use one of the following chassis operational mode commands:

- show chassis hardware
- show chassis fabric fpcs

For more information about chassis operational mode commands, see the *JUNOS System Basics and Services Command Reference*.

Binding a Layer 2 Port-Mirroring Instance to a DPC

You can bind a Layer 2 port-mirroring instance with a specific DPC so that the port-mirroring properties in that instance are applied to all Packet Forwarding Engines (and their associated ports) on that DPC. Port-mirroring properties that are bound to a DPC override the global port-mirroring properties (if the port-mirroring statement has been included at the [edit forwarding-options] hierarchy level).

To bind a named port-mirroring instance to a specific DPC and its Packet Forwarding Engines, include the port-mirror-instance *pm-instance-name* statement at the [edit chassis fpc *slot-number*] hierarchy level.

```
[edit]
chassis {
    fpc slot-number {
```

```
        port-mirror-instance pm-instance-name;
    }
}
```

Binding a Layer 2 Port-Mirroring Instance to a Packet Forwarding Engine

You can bind a Layer 2 port-mirroring instance to a specific Packet Forwarding Engine so that the port-mirroring properties in that instance are applied to all ports associated with that Packet Forwarding Engine. Port-mirroring properties that are bound to a Packet Forwarding Engine override port-mirroring properties bound to the DPC (if the port-mirroring statement has been included at the [edit forwarding-options] hierarchy level).

NOTE: For MX960 routers, there is a one-to-one mapping of Packet Forwarding Engines to Ethernet ports. Therefore, on MX960 routers only, you can configure port-specific bindings of port-mirroring instances.

To associate a port-mirroring instance with a Packet Forwarding Engine and its associated ports, include the port-mirror-instances *pm-instance-name-b* statement at the [edit chassis fpc *slot-number* pic *slot-number*] hierarchy level:

```
[edit]
    fpc slot-number {
        port-mirror-instance pm-instance-name-a;
        pic slot-number {
            port-mirror-instance pm-instance-name-b;
        }
    }
}
```

Precedence of Port-Mirroring Instances at Different Levels of the Chassis

If port-mirroring instances are configured at multiple levels in the MX-series router hierarchy, the port-mirroring properties are applied as follows:

1. **Chassis-level port-mirroring properties apply to all ports in the chassis.** If an MX-series router is configured with the global port-mirroring instance, those chassis-level properties apply to all DPCs and their Packet Forwarding Engines and their associated ports.
2. **FPC-level port-mirroring properties override chassis-level properties.** If a DPC is bound to a named port-mirroring instance, those FPC-level properties apply to all Packet Forwarding Engines (and their associated ports) on the DPC and override the properties bound at the chassis level (if the port-mirroring statement has been included at the [edit forwarding-options] hierarchy level).
3. **PIC-level port-mirroring properties override FPC-level properties.** If a Packet Forwarding Engine is bound to a named port-mirroring instance, those PIC-level port-mirroring properties apply to all ports associated with the Packet Forwarding Engine and override the properties bound at the FPC level (if the port-mirror-instance *pm-instance-name-a* statement has been included at the [edit chassis fpc *slot-number*] hierarchy level).

Layer 2 Port Mirroring for a Logical Interface, Forwarding Table, or Flood Table

You can configure Layer 2 port mirroring by configuring a firewall filter action and then applying the filter at various input or output points in the MX-series system. A Layer 2 port-mirroring firewall filter can be applied to an input or output to a logical interface, including aggregated Ethernet, to an input to a forwarding table for a bridge domain, or to an input to a flood table for a VPLS routing instance:

Configuring a Layer 2 Port-Mirroring Firewall Filter

For the VPLS (family bridge or family vpls) traffic only, MX-series firewall filters can be configured to perform port mirroring if the packet matches the conditions configured in the firewall filter term. A firewall filter configured to perform port mirroring can be applied to input or output logical interfaces, including aggregated Ethernet logical interfaces, or to input to forwarding tables or input to flood tables of bridge domains or VPLS routing instances.

To configure a Layer 2 port-mirroring firewall filter, include the following statements:

```
[edit]
firewall {
    family (bridge | ccc | vpls) {
        filter pm-filter-name {
            term term-name {
                from { # Do not specify match conditions based on route source address
                }
                then {
                    action; # Recommended action is 'accept'
                    port-mirror;
                }
            }
        }
    }
}
```

To configure a firewall filter, include the filter *pm-filter-name* statement at the [edit firewall family (bridge | ccc | vpls)] hierarchy level.

To configure a firewall filter term, include the term *term-name* statement at the [edit firewall family (bridge | ccc | vpls)] filter *pm-filter-name*] hierarchy level.

Under the [edit firewall family (bridge | ccc | vpls)] filter *pm-filter-name* term *term-name*] hierarchy level, do not include the optional from statement that specifies match

conditions based on the route source address. Omit this statement so that all packets are considered to match and all actions specified in the then statement are taken.

To configure the actions to be taken on matching packets, include the then statement under the [edit firewall family (bridge | vpls)] filter *pm-filter-name* term *term-name*] hierarchy level. Within the term, specify an optional *action* and the port-mirror action modifier:

- If you do not specify an action, all input packets are accepted. The recommended action is accept.
- The port-mirror action modifier causes the firewall filter to use the input packet-sampling properties and address family-specific mirror destination properties configured for the Layer 2 port-mirroring global instance of the same family (configured at the [edit forwarding-options port-mirroring] hierarchy level).

Because the port-mirror filter action modifier relies on the global port-mirroring properties, which are configured at the [edit forwarding-options port-mirroring] hierarchy level, the port-mirror filter action is not supported for logical systems.

For detailed information about configuring firewall filters in general (including in a Layer 3 environment), see the *JUNOS Policy Framework Configuration Guide*.

Applying a Layer 2 Port-Mirroring Filter to a Logical Interface

If you apply a Layer 2 port-mirroring firewall filter to a logical interface, only packets received on that logical interface are mirrored. To apply a port-mirroring firewall filter to an input or output logical interface, include the input or output statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (bridge | ccc | vpls) filter] hierarchy level.

- If the filter is to be evaluated when packets are received on the interface, include the input *filter-name* statement.
- If the filter is to be evaluated when packets are sent on the interface, include the output *filter-name* statement.

NOTE: A port-mirroring firewall filter can also be applied to an aggregated-Ethernet logical interface.

```
[edit]
interfaces {
   interface-name {
      vlan-tagging;
      encapsulation extended-vlan-bridge;
      unit number { # Apply a filter to the input of this interface
         vlan-id number;
         family (bridge | ccc | vpls) {
            filter {
               input pm-filter-name-a;
            }
         }
      }
```

```
        unit number { # Apply a filter to the output of this interface
            vlan-id number;
            family (bridge | ccc | vpls) {
                filter {
                    output pm-filter-name-b;
                }
            }
        }
    }
}
```

If port-mirroring firewall filters are applied at both the input and output of a logical interface, two copies of each packet are mirrored. To prevent the router from forwarding duplicate packets to the same destination, include the optional mirror-once statement at the [edit forwarding-options] hierarchy level.

Behavior of a Port-Mirroring Filter Applied to an Aggregated Ethernet Interface

You can apply a Layer 2 port-mirroring firewall filter to an aggregated Ethernet interface to configure port-mirroring at the parent interface. However, if any child interfaces are bound to different Layer 2 port-mirroring instances, packets received at the child interfaces will be mirrored to the destinations specified by their respective port-mirroring instances. Thus, multiple child interfaces can mirror packets to multiple destinations.

For example, suppose the parent aggregated Ethernet interface instance ae0 has two child interfaces:

- xe-2/0/0
- xe-3/1/2

Also suppose that these child interfaces on ae0 are each bound to a different Layer 2 port-mirroring instance:

- pmi-A—Layer 2 port-mirroring instance bound to child interface xe-2/0/0
- pmi-B—Layer 2 port-mirroring instance bound to child interface xe-3/1/2

If you apply a Layer 2 port-mirroring firewall filter to ae0.0 (logical unit 0 on the aggregated Ethernet interface instance 0). This enables port mirroring on ae0.0, which has the following effect on the processing of traffic received on the child interfaces for which Layer 2 port-mirroring properties are specified:

- The packets received on xe-2/0/0.0 are mirrored to the output interfaces configured in port-mirroring instance pmi-A.
- The packets received on xe-3/1/2.0 are mirrored to the output interfaces configured in port-mirroring instance pmi-B.

Because pmi–A and pmi–B might specify different input packet-sampling properties or mirror destination properties, the packets received on xe-2/0/0.0 and xe-3/1/2.0 can mirror different packets to different destinations.

Applying a Layer 2 Port-Mirroring Filter to the Forwarding Table on a Bridge Domain

If a port-mirroring firewall filter is applied to the forwarding table on a bridge domain, any packet received in the bridge domain that matches the filter is mirrored.

To apply a Layer 2 port-mirroring firewall filter to the forwarding table of a bridge domain, include the following statements:

```
bridge-domains {
    bridge-domain-name {
        domain-type bridge;
        interface interface-name;
        forwarding-options {
            filter {
                input pm-filter-name;
            }
        }
    }
}
```

You can include the statements at the following hierarchy levels:

- [edit]
- [edit routing-instances *routing-instance-name*]

Specify the Layer 2 port-mirroring firewall filter in the input *pm-filter-name* statement.

Applying a Layer 2 Port-Mirroring Filter to the Flood Table on a VPLS Routing Instance

If a port-mirroring firewall filter is applied to the flood table on a VPLS routing instance, any packet received in the VPLS routing instance that matches the filter is mirrored.

To apply a Layer 2 port-mirroring firewall filter to the flood table of a VPLS routing instance, include the input *pm-filter-name* statement at the [edit forwarding-options family vpls flood] hierarchy level:

```
[edit]
forwarding-options {
    family vpls {
        flood {
            input pm-filter-name
        }
    }
}
```

Example: Configuring Layer 2 Port Mirroring for a Logical Interface

The following steps describe an example in which the global port-mirroring instance and a port-mirroring firewall filter are used to configure Layer 2 port mirroring for the input to a logical interface.

1. Configure the bridge domain example-bd-with-analyzer, which contains the external packet analyzer, and the bridge domain example-bd-with-traffic, which contains the source and destination of the Layer 2 traffic being mirrored:

```
[edit]
bridge-domains {
   example-bd-with-analyzer { # Contains an external traffic analyzer
      vlan-id 1000;
      interface ge-2/0/0.0; # External analyzer
   }
   example-bd-with-traffic { # Contains traffic input and output interfaces
      vlan-id 1000;
      interface ge-2/0/6.0; # Traffic input port
      interface ge-3/0/1.2; # Traffic output port
   }
}
```

Assume that logical interface ge-2/0/0.0 is associated with an external traffic analyzer that is to receive port-mirrored packets. Assume that logical interfaces ge-2/0/6.0 and ge-3/0/1.2 will be traffic input and output ports, respectively.

2. Configure Layer 2 port-mirroring for the global instance, with the port-mirroring destination being the bridge domain interface associated with the external analyzer (logical interface ge-2/0/0.0 on bridge domain example-bd-with-analyzer). Be sure to enable the option that allows filters to be applied to this port-mirroring destination:

```
[edit]
forwarding-options {
   port-mirroring {
      input {
         rate 10;
         run-length 5;
      }
      family bridge {
         output {
            interface ge-2/0/0.0; # Mirror packets to the external analyzer
            no-filter-check; # Allow filters on the mirror destination interface
         }
      }
   }
}
```

The input statement under the [edit forwarding-options port-mirroring] hierarchy level specifies that sampling begins every tenth packet and that each of the first five packets sampled are to be mirrored.

The output statement under the [edit forwarding-options port-mirroring family bridge] hierarchy level specifies the output mirror interface for Layer 2 packets in a bridging environment:

- Logical interface ge-2/0/0.0, which is associated with the external packet analyzer, is configured as the port-mirroring destination.
- The optional no-filter-check statement allows filters to be configured on this destination interface.

3. Configure the Layer 2 port-mirroring firewall filter example-bridge-pm-filter:

```
firewall {
```

```
    family bridge {
        filter example-bridge-pm-filter {
            term example-filter-terms {
                then {
                    accept;
                    port-mirror;
                }
            }
        }
    }
}
```

When this firewall filter is applied to the input or output of a logical interface for traffic in a bridging environment, Layer 2 port mirroring is performed according to the input packet-sampling properties and mirror destination properties configured for the Layer 2 port mirroring global instance. Because this firewall filter is configured with the single, default filter action **accept**, all packets selected by the **input** properties (**rate = 10** and **run-length = 5**) match this filter.

4. Configure the logical interfaces:

```
[edit]
interfaces {
    ge-2/0/0 { # Define the interface to the external analyzer
        encapsulation ethernet-bridge;
        unit 0 {
            family bridge;
        }
    }
    ge-2/0/6 { # Define the traffic input port
        flexible-vlan-tagging;
        encapsulation extended-vlan-bridge;
        unit 0 {
            vlan-id 100;
            family bridge {
                filter {
                    input example-bridge-pm-filter; # Apply the port-mirroring firewall filter
                }
            }
        }
    }
    ge-3/0/1 { # Define the traffic output port
        flexible-vlan-tagging;
        encapsulation extended-vlan-bridge;
        unit 2 {
            vlan-tags outer 10 inner 20;
            family bridge;
        }
    }
}
```

Packets received at logical interface ge-2/0/6.0 on bridge domain example-bd-with-traffic are evaluated by the port-mirroring firewall filter example-bridge-pm-filter. The firewall filter acts on the input traffic according to the filter actions configured in the firewall filter itself plus the input packet-sampling properties and mirror destination properties configured in the global port-mirroring instance:

- All packets received at ge-2/0/6.0 are forwarded to their (assumed) normal destination at logical interface ge-3/0/1.2.
- For every ten input packets, copies of the first five packets in that sample are forwarded to the external analyzer at logical interface ge-0/0/0.0 in the other bridge domain, example-bd-with-analyzer.

If you configure the port-mirroring firewall filter example-bridge-pm-filter to take the discard action instead of the accept action, all original packets are discarded while copies of the packets selected using the global port-mirroring input properties are sent to the external analyzer.

Example: Configuring Layer 2 Port Mirroring on an L2VPN

The following example is not a complete configuration, but shows all the steps needed to configure port mirroring on an L2VPN using family ccc.

1. Configure the bridge domain port-mirror-bd, which contains the external packet analyzer:

```
[edit]
bridge-domains {
   port-mirror-bd { # Contains an external traffic analyzer
      interface ge-2/2/9.0; # External analyzer
   }
}
```

2. Configure the Layer 2 VPN CCC to connect interface ge-2/0/1.0 and interface ge-2/0/1.1:

```
[edit]
   protocols {
   mpls {
      interface all;
   }
   connections {
      interface-switch if_switch {
         interface ge-2/0/1.0;
         interface ge-2/0/1.1;
      }
   }
}
```

3. Configure Layer 2 port mirroring for the global instance, with the port-mirroring destination being the bridge domain interface associated with the external analyzer (logical interface ge-2/2/9.0 on bridge domain example-bd-with-analyzer):

```
[edit]
forwarding-options {
   port-mirroring {
      input {
         rate 1;
         maximum-packet-length 200;
      }
      family ccc {
         output {
            interface ge-2/2/9.0; # Mirror packets to the external analyzer
         }
      }
      instance {
         inst1 {
            input {
               rate 1;
               maximum-packet-length 300;
            }
            family ccc {
               output {
                  interface ge-2/2/9.0;
               }
            {
         }
      }
   }
}
```

4. Configure for firewall filter **pm-ccc** for **family ccc**:

```
[edit]
firewall {
   family ccc {
      filter pm_ccc {
         term pm {
            then port-mirror;
         }
      }
   }
}
```

5. Apply the port mirror instance to the chassis:

```
[edit]
chassis {
   fpc 2 {
      port-mirror-instance inst1;
   }
}
```

6. Configure interfaces **ge-2/0/1** (for the VLANs) and **ge-2/2/9** (for port mirroring) with the **pm-ccc** filter:

```
[edit]
interfaces {
   ge-2/0/1 {
```

```
        vlan-tagging;
        encapsulation extended-vlan-ccc;
        unit 0 {
            vlan-id 10;
            family ccc {
                filter {
                    input pm_ccc;
                }
            }
        }
        unit 1 {
            vlan-id 20;
            family ccc {
                filter {
                    output pm_ccc;
                }
            }
        }
    }
    ge-2/2/9 {
        encapsulation ethernet-bridge;
        unit 0 {
            family bridge;
        }
    }
}
```

Example: Configuring Layer 2 Port Mirroring on an L2VPN with AE

The following example is not a complete configuration, but shows all the steps needed to configure port mirroring on an L2VPN using family ccc and aggregated Ethernet (AE) links.

1. Configure the bridge domain port-mirror-bd, which contains the external packet analyzer:

```
[edit]
bridge-domains {
    port-mirror-bd { # Contains an external traffic analyzer
        interface ge-2/2/8.0; # External analyzer
    }
}
```

2. Configure the Layer 2 VPN CCC to connect interface ae0.0 and interface ae0.1:

```
[edit]
    protocols {
    mpls {
        interface all;
    }
    connections {
        interface-switch if_switch {
            interface ae0.0;
            interface ae0.1;
```

```
        }
    }
}
```

3. Configure Layer 2 port mirroring for the global instance, with the port-mirroring destination being the bridge domain interface associated with the external analyzer (logical interface ge-2/2/9.0 on bridge domain example-bd-with-analyzer):

```
[edit]
forwarding-options {
    port-mirroring {
        input {
            rate 1;
            maximum-packet-length 200;
        }
        family ccc {
            output {
                interface ge-2/2/8.0; # Mirror packets to the external analyzer
            }
        }
        instance {
            inst1 {
                input {
                    rate 1;
                    maximum-packet-length 300;
                }
                family ccc {
                    output {
                        interface ge-2/2/8.0;
                    }
                {
            }
        }
    }
}
```

4. Configure for firewall filter pm-ccc for family ccc:

```
[edit]
firewall {
    family ccc {
        filter pm_ccc {
            term pm {
                then port-mirror;
            }
        }
    }
}
```

5. Apply the aggregated Ethernet interfaces and port mirror instance to the chassis:

```
[edit]
chassis {
    aggregaated-devices {
        ethernet {
            device-count 10;
```

```
    }
  }
  fpc 2 {
    port-mirror-instance inst1;
  }
}
```

6. Configure interfaces ae0 and ge-2/0/2 (for aggregated Ethernet) and ge-2/2/8 (for port mirroring) with the pm_ccc filter:

```
[edit]
interfaces {
  ae0 {
    vlan-tagging;
    encapsulation extended-vlan-ccc;
    unit 0 {
      vlan-id 10;
      family ccc {
        filter {
          input pm_ccc;
        }
      }
    }
    unit 1 {
      vlan-id 20;
      family ccc {
        filter {
          output pm_ccc;
        }
      }
    }
  }
  ge-2/0/2 {
    gigether-options {
      802.3ad ae0;
    }
  }
  ge-2/2/8 {
    encapsulation ethernet-bridge;
    unit 0 {
      family bridge;
    }
  }
}
```

Part 3
Layer 2 Bridging

Chapter 4

Configuring Layer 2 Bridging

This chapter describes how 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 (STP) instance and separate its VLAN ID space.

Layer 2 Bridging Overview

On MX-series routers only, you can configure one or more bridge domains to perform Layer 2 bridging. 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. 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 STP instance and separate its VLAN ID space. Thus, each virtual switch can participate in a different Layer 2 network.

Beginning with JUNOS Release 9.2, 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.

Beginning with JUNOS Release 9.3, 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.

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. For more detailed information about how to configure IRB for a bridge domain, see “Configuring Integrated Routing and Bridging for a Bridge Domain” on page 43. To enable a bridge domain, include the following statements:

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

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 a Bridge Domain or a VPLS Routing Instance” on page 39.

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. For more information about how to configure Layer 2 forwarding properties for a bridge domain, see "Configuring Layer 2 Learning and Forwarding Properties for a Bridge Domain" on page 51. For more information about how to configure Layer 2 forwarding properties for a set of bridge domains with a Layer 2 trunk port, see "Configuring Layer 2 Learning and Forwarding Properties for a Set of Bridge Domains with a Layer 2 Trunk Port" on page 56. You can also configure Layer 2 address learning and forwarding properties for an MX-series router as a whole. For more information, see "Configuring Layer 2 Address Learning and Forwarding Properties" on page 77.

You can also configure spanning-tree protocols to prevent forwarding loops at the [edit protocols mstp] hierarchy level. For more information, see "Configuring Spanning-Tree Protocols" on page 87.

Beginning with JUNOS Release 8.5, you can configure IGMP snooping for a bridge domain. For more information, see the *JUNOS Multicast Protocols Configuration Guide*.

Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance

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 Network Interfaces Configuration Guide*.
- 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*] or [edit logical-systems *logical-system-name* interfaces *interface-name*] 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 38.

For a VPLS routing instance, include either the vlan-id or vlan-tags statement at the [edit interfaces *interface-name*] or [edit logical-systems *logical-system-name* interfaces *interface-name*] 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 VPNs Configuration Guide*.

NOTE: For a single bridge domain or VPLS routing instance, you can configure either the vlan-id statement or the vlan-tags statement, but not both.

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 7 on page 42. 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 8 on page 42.

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

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 7 on page 42.
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 8 on page 42.

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 7 on page 42 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 7: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain

	VLAN Configurations for Bridge Domain			
VLAN Identifier of Logical Interface	**vlan-id none**	**vlan-id 200**	**vlan-id all**	**vlan tags outer 100 inner 300**
none	No operation	push 200	–	push 100, push 300
200	pop 200	No operation	No operation	swap 200 to 300, push 100
1000	pop 1000	swap 1000 to 200	No operation	swap 1000 to 300, push 100
vlan-tags outer 2000 inner 300	pop 2000, pop 300	pop 2000, swap 300 to 200	pop 2000	swap 2000 to 100
vlan-tags outer 100 inner 400	pop 100, pop 400	pop 100, swap 400 to 200	pop 100	swap 400 to 300
vlan-id-range 10–100	–	–	No operation	–
vlan-tags outer 200 inner-range 10–100	–	–	pop 200	–

Table 8 on page 42 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 8: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain

	VLAN Configurations for Bridge Domain			
VLAN Identifier of Logical Interface	**vlan-id none**	**vlan-id 200**	**vlan-id all**	**vlan tags outer 100 inner 300**
none	no operation	pop 200	–	pop 100, pop 300

Table 8: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain *(continued)*

VLAN Identifier of Logical Interface	VLAN Configurations for Bridge Domain			
	vlan-id none	**vlan-id 200**	**vlan-id all**	**vlan tags outer 100 inner 300**
200	push 200	No operation	No operation	pop 100, swap 300 to 200
1000	push 1000	swap 200 to 1000	No operation	pop 100, swap 300 to 1000
vlan-tags outer 2000 inner 300	push 2000, push 300	swap 200 to 300, push 2000	push 2000	swap 100 to 2000
vlan-tags outer 100 inner 400	push 100, push 400	swap 200 to 400, push 100	push 100	swap 300 to 400
vla-id-range 10–100	–	–	No operation	–
vlan-tags outer 200 inner-range 10–100	–	–	push 200	–

Configuring Integrated Routing and Bridging for a Bridge Domain

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 Network Interfaces Configuration Guide*.

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

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

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

For each bridge domain that you configure, specify a *bridge-domain-name*. You must also specify bridge as the domain-type.

For the vlan-id statement, you can specify either a valid VLAN identifier or the none option. For more information about configuring VLAN identifiers, see "Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance" on page 39.

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 configure 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 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 Network Interfaces Configuration Guide*.

Beginning with JUNOS Release 9.0, IRB interfaces are supported for multicast snooping. For more information about multicast snooping, see the *JUNOS Multicast Protocols Configuration Guide*.

Configuring a Set of Bridge Domains for a Layer 2 Trunk Port

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:

```
[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 [ numbers ];
        }
    }
}
[edit bridge-domains]
bridge-domain-name {
    vlan-id number;
    . . . .
}
```

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 *JUNOS Network Interfaces Configuration Guide*.

Configuring Layer 2 Virtual Switches

On the 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 (STP) 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, or Multiple Spanning Tree Protocol—to prevent forwarding loops. For more information about Layer 2 control protocols, see “Configuring Spanning-Tree Protocols” on page 87. For more information about how to configure Layer 2 logical ports on an interface, see the *JUNOS Network Interfaces Configuration Guide*.

Beginning with JUNOS Release 9.2, 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 Network Interfaces Configuration Guide*.

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 Layer 2 forwarding and learning properties for a bridge

domain, see “Configuring Layer 2 Learning and Forwarding Properties for a Bridge Domain” on page 51.

For more information about configuring a routing instance for Layer 2 VPN, see the *JUNOS VPNs Configuration Guide*. For a detailed Layer 2 VPN example configuration, see the *JUNOS Feature Guide*.

For information about configuring Layer 2 protocol tunneling, see “Configuring Layer 2 Protocol Tunneling” on page 100.

For more information about how to configure Layer 2 routing instances, see the following sections:

- Configuring a Layer 2 Virtual Switch on page 46
- Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port on page 47
- Configuring VPLS Ports in a Virtual Switch on page 48
- Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch on page 50

Configuring a Layer 2 Virtual Switch

A Layer 2 virtual switch, which isolates a LAN segment with its Spanning Tree Protocol (STP) 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:

```
[edit]
routing-instances {
   routing-instance-name (
      instance-type virtual-switch;
      bridge-domains {
```

```
        bridge-domain-name {
            domain-type bridge;
            vlan-id (all | none | number); # Cannot be used with 'vlan-tags' statement
            vlan-tags outer number inner number; # Cannot be used with 'vlan-id'
                statement
            interface interface-name;
        }
    }
    protocols {
        mstp ...
    }
  }
}
```

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 bridge as the domain-type.

For the vlan-id statement, you can specify either a valid VLAN identifier or the none or all options. If you specify a valid VLAN identifier, you cannot also use the none option. These statements are mutually exclusive. For more information about configuring VLAN identifiers and VLAN tags for a bridge domain, see "Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance" on page 39.

The all option is not supported with IRB. For more information about how to configure IRB, see "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

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 configure either the vlan-id statement or the vlan-tags statement, but not both.

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 Network Interfaces Configuration Guide*.

For information about how to configure spanning-tree protocols, see the *JUNOS Feature Guide*.

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

You can associate one or more Layer 2 trunk interfaces with a virtual switch. A 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 Network Interfaces Configuration Guide*.

In addition, you can configure Layer 2 learning and forwarding properties for the virtual switch. For more information, see "Configuring Layer 2 Learning and Forwarding Properties for a Set of Bridge Domains with a Layer 2 Trunk Port" on page 56.

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.

Configuring VPLS Ports in a Virtual Switch

Beginning with JUNOS Release 9.3, 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:

```
[edit]
interfaces {
   interface-name {
      unit logical-unit-number { # Call this 'L2-trunk-port-A'
         family bridge {
            interface-mode trunk;
            vlan-id-list [ 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 [ 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 [*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; #
            interface L2-trunk-port-A;
         }
         bridge-domain-name-1 {
            domain-type bridge;
```

```
            vlan-id number;
            interface L2-trunk-port-B;
          }
        }
        protocols {
          vpls {
            vlan-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 *number*, and interface *interface-name-trunk-port* statements at the [edit routing-instances *routing-instance-name* bridge-domain *bridge-domain-name*] hierarchy level.

Configuring Integrated Routing and Bridging for a Bridge Domain Within 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 Network Interfaces Configuration Guide*.

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 {
  routing-instance-name {
    instance-type virtual switch;
    bridge-domains {
      bridge-domain-name {
        domain-type bridge;
```

```
            vlan-id (none | number);
            vlan-tags outer number inner number;
            interface interface-name;
            routing-interface routing-interface-name;
        }
    }
  }
}
```

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 bridge as the domain-type.

For the vlan-id statement, you can specify either a valid VLAN identifier or the none option. For more information about configuring VLAN identifiers, see "Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance" on page 39.

NOTE: For a single bridge domain, you can configure 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 Network Interfaces Configuration Guide*.

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.

Configuring Layer 2 Learning and Forwarding Properties for a Bridge Domain

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

For more information about how to configure Layer 2 learning and forwarding properties for an MX-series router, see "Configuring Layer 2 Address Learning and Forwarding Properties" on page 77.

For more information about how to configure Layer 2 learning and forwarding properties for a bridge domain, see the following sections:

- Disabling MAC Learning for a Bridge Domain or Logical Interface on page 52
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 53
- Configuring the Size of the MAC Address Table on page 54
- Limiting the Number of MAC Addresses Learned from an Interface in a Bridge Domain on page 54
- Enabling MAC Accounting for a Bridge Domain on page 56

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. You can also disable MAC learning for an MX-series router. For more information, see "Disabling MAC Learning" on page 77.

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.

For more information about how to disable MAC learning for the entire MX-series router, see "Disabling MAC Learning" on page 77.

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.

Configuring the Size of the MAC Address Table

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;
      }
   }
}
```

Limiting the Number of 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, include the interface-mac-limit *limit* 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
```

```
            interface-mac-limit limit;
        }
    }
}
```

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 16 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:

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

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:

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

You can also configure a limit to the number of MAC addresses learned for an MX-series router. For more information, see “Limiting the Number of MAC Addresses Learned from Each Interface” on page 79.

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 software 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;
    }
}
```

Configuring Layer 2 Learning and Forwarding Properties for a Set of Bridge Domains with a Layer 2 Trunk Port

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

For more information about how to configure Layer 2 learning and forwarding properties for a set of bridge domains, see the following sections:

- Disabling MAC Learning for a Set of Bridge Domains on page 56
- Limiting the Number of MAC Addresses Learned from a Trunk Port on page 57
- Modifying the Size of the MAC Address Table for a Set of Bridge Domains on page 58
- Enabling MAC Accounting for a Set of Bridge Domains on page 58

Disabling MAC Learning 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;
```

Limiting the Number of MAC Addresses Learned from a 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 16 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;
```

Modifying 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 *limit* statement at the [edit switch-options] hierarchy level:

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

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 software 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;
```

Configuring Layer 3 Tunnel Services Interfaces on MX-series Routers

The MX-series routers support Dense Port Concentrators (DPCs) with built-in Ethernet ports and therefore do not support Tunnel Services PICs. To create tunnel interfaces on an MX-series router, you configure a DPC and the corresponding Packet Forwarding Engine to use for tunneling services at the [edit chassis] hierarchy level. You also configure the amount of bandwidth reserved for tunnel services. The JUNOS software creates tunnel interfaces on the Packet Forwarding Engine. To create tunnel interfaces on MX-series routers, include the following statements at the [edit chassis] hierarchy level:

```
[edit chassis]
fcp slot-number {
    pic number {
        tunnel-services {
            bandwidth (1g | 10g);
        }
    }
}
```

Include the fpc *slot-number* statement to specify the slot number of the DPC. If two SCBs are installed, the range is 0 through 11. If three SCBs are installed, the range is 0 through 5 and 7 through 11.

Include the pic *number* statement to specify the number of the Packet Forwarding Engine on the DPC. The range is 0 through 3.

You can also specify the amount of bandwidth to allocate for tunnel traffic on each Packet Forwarding Engine by including the bandwidth (1g | 10g) statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level:

- 1g indicates that 1 Gbps of bandwidth is reserved for tunnel traffic. Configure this option only for a Packet Forwarding Engine on a Gigabit Ethernet 40-port DPC.
- 10g indicates that 10 Gbps of bandwidth is reserved for tunnel traffic. Configure this option only for a Packet Forwarding Engine on a 10-Gigabit Ethernet 4-port DPC.

If you specify a bandwidth that is not compatible with the type of DPC and Packet Forwarding Engine, tunnel services are not activated. For example, you cannot specify a bandwidth of 1 Gbps for a Packet Forwarding Engine on a 10-Gigabit Ethernet 4-port DPC.

When you configure tunnel interfaces on the Packet Forwarding Engine of a 10-Gigabit Ethernet 4-port DPC, the Ethernet interfaces for that port are removed from service and are no longer visible in the command-line interface (CLI). The Packet Forwarding Engine of a 10-Gigabit Ethernet 4-port DPC supports either tunnel interfaces or Ethernet interfaces, but not both. Each port on the 10-Gigabit Ethernet 4-port DPC includes two LEDs, one for tunnel services and one for Ethernet services, to indicate which type of service is being used. On the Gigabit Ethernet 40-port DPC, you can configure both tunnel and Ethernet interfaces at the same time.

To verify that the tunnel interfaces have been created, issue the show interfaces terse operational mode command. For more information, see the *JUNOS Interfaces Command Reference*.

For additional information about tunnel services, see the “Tunnel Services” chapter in the *JUNOS Services Interfaces Configuration Guide*.

Chapter 5

Summary of Bridge Domain Configuration Statements

The following sections explain each of the bridge domain configuration statements. The statements are organized alphabetically.

bandwidth

Syntax bandwidth (1g | 10g);

Hierarchy Level [edit chassis fpc *slot-number* pic *number* tunnel-services]

Release Information Statement introduced in JUNOS Release 8.2.

Description On the MX-series Ethernet Services routers only, specify the amount of bandwidth to reserve for tunnel services.

Options 1g—Specify a bandwidth of 1 Gbps on the Packet Forwarding Engine connected to a Gigabit Ethernet 40-port Dense Port Concentrator (DPC).

10g—Specify a bandwidth of 10 Gbps on the Packet Forwarding Engine connected to 10-Gigabit Ethernet 4-port DPC.

NOTE: If you specify a bandwidth that is not compatible with the type of DPC and Packet Forwarding Engine, tunnel services are not activated. For example, you cannot specify a bandwidth of 1 GPS for a Packet Forwarding Engine on a 10-Gigabit Ethernet 4-port DPC.

Usage Guidelines See "Configuring Layer 3 Tunnel Services Interfaces on MX-series Routers" on page 58.

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

bridge-domains

Syntax

```
bridge-domains bridge-domain-name {
    domain-type bridge;
    vlan-id (all | none | number);
    vlan-tags outer number inner number;
    routing-interface routing-interface-name;
    interface interface-name;
    bridge-options {
        interface-mac-limit limit;
        mac-statistics;
        mac-table-size limit;
        no-mac-learning;
        interface interface-name;
            static-mac mac-address;
        }
    }
}
```

Hierarchy Level [edit],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced in JUNOS Release 8.4.

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.

Options *bridge-domain-name*—Name of the bridge domain.

The remaining statements are explained separately.

Usage Guidelines See "Configuring a Bridge Domain" on page 38, "Configuring a Layer 2 Virtual Switch" on page 46, and "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

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

Related Topics instance-type

bridge-options

Syntax

```
bridge-options {
    interface-mac-limit limit;
        packet-action;
    }
    mac-statistics;
    mac-table-size limit;
    no-mac-learning;
    interface interface-name;
        static-mac static-mac-address;
    }
}
```

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domain *bridge-domain-name*]

Release Information Statement introduced in JUNOS Release 8.4.

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.

Usage Guidelines See "Configuring Layer 2 Learning and Forwarding Properties for a Bridge Domain" on page 51.

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

Related Topics l2-learning, switch-options

domain-type

Syntax domain-type bridge;

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description (MX-series routers only) Define the type of domain for a Layer 2 bridge domain.

Usage Guidelines See "Configuring a Bridge Domain" on page 38, "Configuring a Layer 2 Virtual Switch" on page 46, and "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

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

interface

Syntax interface *interface-name*;

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced in JUNOS Release 8.4.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2.
Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Description (MX-series routers only) Specify the logical interfaces to include in the bridge domain, 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 Network Interfaces Configuration Guide*.

Usage Guidelines See "Configuring a Bridge Domain" on page 38, "Configuring a Layer 2 Virtual Switch" on page 46, "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50, and "Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port" on page 47.

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

Related Topics routing-interface

interface-mac-limit

Syntax

```
interface-mac-limit limit {
    packet-action drop;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* bridge-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* switch-options],
[edit routing-instances *routing-instance-name* switch-options interface *interface-name*],
[edit switch-options],
[edit switch-options interface *interface-name*]

Release Information

Statement introduced in JUNOS Release 8.4.
Support for the switch-options statement introduced in JUNOS Release 9.2.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2. Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Description

(MX-series routers only) Configure a limit to the number of MAC addresses that can be learned from a bridge domain, virtual switch, or set of bridge domains.

Default

1024 MAC addresses for each logical interface.

Options

limit—Maximum number of MAC addresses learned from an interface.

Range: 16 through 131,071 MAC addresses per interface

The remaining statement is explained separately.

Usage Guidelines

See “Limiting the Number of MAC Addresses Learned from an Interface in a Bridge Domain” on page 54 and “Limiting the Number of MAC Addresses Learned from a Trunk Port” on page 57.

Required Privilege Level

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

Related Topics

global-mac-limit and *JUNOS VPNs Configuration Guide*

mac-statistics

Syntax mac-statistics;

Hierarchy Level [edit bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* switch-options],
[edit switch-options]

Release Information Statement introduced in JUNOS Release 8.4.
Support for the switch-options statement introduced in JUNOS Release 9.2.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2. Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Description (MX-series routers only) Enable MAC accounting either for a specific bridge domain, or for a set of bridge domains associated with a Layer 2 trunk port.

Default disabled

Usage Guidelines See "Enabling MAC Accounting for a Bridge Domain" on page 56 and "Enabling MAC Accounting for a Set of Bridge Domains" on page 58.

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

Related Topics global-mac-statistics

mac-table-size

Syntax

```
mac-table-size limit;
    packet-action drop;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* switch-options],
[edit switch-options]

Release Information

Statement introduced in JUNOS Release 8.4.
Support for the switch-options statement introduced in JUNOS Release 9.2.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2. Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Description

Modify the size of the MAC address table for the bridge domain, a set of bridge domains associated with a trunk port, or a virtual switch. The default is 5120 MAC addresses.

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

The remaining statement is explained separately.

Usage Guidelines

See “Configuring the Size of the MAC Address Table” on page 54 and “Modifying the Size of the MAC Address Table for a Set of Bridge Domains” on page 58.

Required Privilege Level

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

Related Topics

interface-mac-limit and *JUNOS VPNs Configuration Guide*

no-mac-learning

Syntax no-mac-learning;

Hierarchy Level
[edit bridge-domain *bridge-domain-name* bridge-options],
[edit bridge-domain *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* switch-options],
[edit switch-options]

Release Information
Statement introduced in JUNOS Release 8.4.
Support for the switch-options statement introduced in JUNOS Release 9.2.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2. Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or bridge domain configured within a virtual switch.

Description (MX-series routers only) Disable MAC learning for a virtual switch, for a bridge domain, for a specific logical interface in a bridge domain, or for a set of bridge domains associated with a Layer 2 trunk port.

Default MAC learning is enabled. Use no-mac-learning to disable MAC learning.

Usage Guidelines See and “Disabling MAC Learning for a Bridge Domain or Logical Interface” on page 52 and “Disabling MAC Learning for a Set of Bridge Domains” on page 56.

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

Related Topics global-no-mac-learning

packet-action

Syntax packet-action drop;

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 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* switch-options interface *interface-name* interface-mac-limit *limit*],
[edit routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],
[edit switch-options interface-mac-limit *limit*]

Release Information Statement introduced in JUNOS Release 8.4.
Support for the switch-options statement introduced in JUNOS Release 9.2.
Support for top-level configuration for the virtual-switch type of routing instance introduced in JUNOS Release 9.2. Before JUNOS Release 9.2, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Description (MX-series routers only) Specify that packets for new source MAC addresses be dropped after the MAC address limit is reached.

Default Disabled. The default is for packets for new source MAC addresses to be forwarded after the MAC address limit is reached.

Usage Guidelines See “Limiting the Number of MAC Addresses Learned from an Interface in a Bridge Domain” on page 54 and “Limiting the Number of MAC Addresses Learned from a Trunk Port” on page 57.

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

Related Topics interface-mac-limit and *JUNOS VPNs Configuration Guide*

routing-interface

Syntax routing-interface *routing-interface-name*;

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domains-name*]

Release Information Statement introduced in JUNOS Release 8.4.

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

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 Network Interfaces Configuration Guide*.

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

Usage Guidelines See "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50 and "Configuring a Bridge Domain" on page 38.

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

Related Topics interface and *JUNOS VPNs Configuration Guide*

static-mac

Syntax

```
static-mac mac-address {
    <vlan-id number>;
}
```

Hierarchy Level [edit 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*]

Release Information Statement introduced in JUNOS Release 8.4.

Description (MX-series routers only) Configure a static MAC address for a logical interface in a bridge domain.

Options *mac-address*—MAC address

vlan-id *number*—(Optional) VLAN identifier to associate with static MAC address.

Usage Guidelines See "Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain" on page 53.

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

switch-options

Syntax
```
switch-options {
    interface-mac-limit limit {
        packet-action drop;
    }
    mac-statistics;
    mac-table-size size;
    no-mac-learning;
    interface interface-name {
        interface-mac-limit limit;
    }
}
```

Hierarchy Level [edit],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced in JUNOS Release 9.2.

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

Options The remaining statements are explained separately.

Usage Guidelines See “Configuring Layer 2 Learning and Forwarding Properties for a Set of Bridge Domains with a Layer 2 Trunk Port” on page 56.

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

Related Topics bridge-options, l2-learning

tunnel-services

Syntax
```
tunnel-services {
bandwidthtunnel-services (1g | 10g);
```

Hierarchy Level [edit chassis fpc *slot-number* pic *number*]

Release Information Statement introduced in JUNOS Release 8.2.

Description For MX-series Ethernet Services Routers, configure the amount of bandwidth for tunnel services.

Options The statements are explained separately.

Usage Guidelines See “Configuring Layer 3 Tunnel Services Interfaces on MX-series Routers” on page 58.

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

vlan-id

Syntax vlan-id (all | none | *number*);

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

Release Information Statement introduced in JUNOS Release 8.4.
Support for Layer 2 trunk port introduced in JUNOS Release 9.2.

Description (MX-series routers only) Specify a VLAN identifier to include in the packets sent to and from the bridge domain or a VPLS routing instance.

Options *number*—A valid VLAN identifier. If you configure multiple bridge domains with a valid VLAN identifier, you must specify a unique VLAN identifier for each domain. However, you can use the same VLAN identifier for bridge domains that belong to different virtual switches. Use this option to send singly tagged frames with the specified VLAN identifier over VPLS VT interfaces.

NOTE: If you specify a VLAN identifier, you cannot also use the all option. They are mutually exclusive.

all—Specify that the bridge domain spans all the VLAN identifiers configured on the member logical interfaces.

NOTE: You cannot specify the all option if you include a routing interface in the bridge domain.

none—Specify to enable shared VLAN learning or to send untagged frames over VPLS VT interfaces.

Usage Guidelines See "Configuring a Bridge Domain" on page 38, "Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance" on page 39, "Configuring a Set of Bridge Domains for a Layer 2 Trunk Port" on page 44, "Configuring a Layer 2 Virtual Switch" on page 46, "Configuring VPLS Ports in a Virtual Switch" on page 48, and "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

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

Related Topics vlan-tags and *JUNOS VPNs Configuration Guide*

vlan-tags

Syntax vlan-tags outer *number* inner *number*;

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description (MX-series routers only) Specify dual VLAN identifier tags for a bridge domain or a VPLS routing instance.

Options outer *number*—A valid VLAN identifier.

inner *number*—A valid VLAN identifier.

Usage Guidelines See "Configuring a Bridge Domain" on page 38, "Configuring VLAN Identifiers for a Bridge Domain or a VPLS Routing Instance" on page 39, "Configuring a Layer 2 Virtual Switch" on page 46, and "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

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

Related Topics vlan-id and *JUNOS Network Interfaces Configuration Guide*

Part 4

Layer 2 Address Learning and Forwarding

Chapter 6

Configuring Layer 2 Address Learning and Forwarding Properties

This chapter describes how you can configure Layer 2 MAC address and VLAN learning and forwarding on the MX-series routers to support Layer 2 bridging.

Layer 2 Address Learning and Forwarding Properties Overview

On MX-series routers only, you can configure Layer 2 address 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 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.

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

For more information about how to configure bridge domains and virtual switches, see “Configuring Layer 2 Bridging” on page 37 and “Configuring Layer 2 Virtual Switches” on page 45.

Disabling MAC Learning

Disabling dynamic MAC learning on an MX-series router prevents all the logical interfaces on the router from learning source and destination MAC addresses.

To disable MAC learning for an MX-series router, include the global-no-mac-learning statement at the [edit protocols l2-learning] hierarchy level:

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

For more information about how to disable MAC learning for a bridge domain or a specific logical interface, see "Disabling MAC Learning for a Bridge Domain or Logical Interface" on page 52. For more information about how to configure a virtual switch, see "Configuring a Layer 2 Virtual Switch" on page 46 and "Configuring Integrated Routing and Bridging for a Bridge Domain Within a Layer 2 Virtual Switch" on page 50.

Configuring the MAC Table Timeout Interval

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 only for specific MAC table entries, such as for a bridge domain or 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. For more information about configuring static MAC addresses, see "Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain" on page 53.

To modify the timeout interval for the MAC table for the entire routing platform, include the global-mac-table-aging-time *seconds* statement at the [edit protocols l2-learning] hierarchy level:

```
[edit protocols l2-learning]
global-mac-table-aging-time seconds;
```

The range for *seconds* is from 10 through 1,000,0000.

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 software 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 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:

```
[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:

```
[edit protocols l2-learning global-mac-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. For more information, see “Limiting the Number of MAC Addresses Learned from an Interface in a Bridge Domain” on page 54.

NOTE: Starting in JUNOS Release 8.4 on MX-series routers, 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.

Configuring MAC Move Parameters

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

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

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

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

Use the **show l2-learning mac-move-buffer** command to view detailed information about MAC address moves.

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

```
[edit l2–learing]
[Unresolved xref] {
    notificationPH;
    thrsehoildPH;
    countPH;
}
```

Chapter 7

Summary of Layer 2 Address Learning and Forwarding Configuration Statements

The following sections explain each of the Layer 2 address learning and forwarding configuration statements. These statements are organized alphabetically.

global-mac-limit

Syntax

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

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in JUNOS Release 8.4.

Description (MX-series routers only) Limit the number of media access control (MAC) addresses learned from the logical interfaces on the router.

Default 393,215 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 in the "Summary of Bridge Domain Configuration Statements" chapter.

Usage Guidelines See "Limiting the Number of MAC Addresses Learned from Each Interface" on page 79.

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

Related Topics interface-mac-limit

global-mac-statistics

Syntax global-mac-statistics;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in JUNOS Release 9.2.

Description (MX-series routers only) Enable MAC accounting for the entire router,

Default disabled

Usage Guidelines See "Enabling MAC Accounting" on page 78.

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

Related Topics mac-statistics

global-mac-table-aging-time

Syntax global-mac-table-aging-time *seconds*;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in JUNOS Release 9.2.

Description (MX-series routers only) 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: 10 through 1 million

Usage Guidelines See "Configuring the MAC Table Timeout Interval" on page 78.

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

Related Topics *JUNOS VPNs Configuration Guide*

global-no-mac-learning

Syntax global-no-mac-learning;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in JUNOS Release 9.2.

Description (MX-series routers only) Disable MAC learning for the entire router.

Default MAC learning is enabled.

Usage Guidelines See "Disabling MAC Learning" on page 77.

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

Related Topics no-mac-learning

l2-learning

Syntax
```
l2-learning {
    global-mac-limit limit;
    global-mac-statistics;
    global-mac-table-aging-time seconds;
    global-no-mac-learning;
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in JUNOS Release 8.4.

Description (MX-series routers only) Configure Layer 2 address learning and forwarding properties globally.

The statements are explained separately.

Usage Guidelines See "Configuring Layer 2 Address Learning and Forwarding Properties" on page 77.

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

Related Topics bridge-options, switch-options

Part 5
Spanning Tree Protocols

Chapter 8

Configuring Spanning-Tree Protocols

This chapter describes how you can configure the various versions of the Spanning Tree Protocol (STP) supported on MX-series routers to create a loop-free topology in Layer 2 networks.

Spanning-Tree Protocols Overview

The Spanning Tree Protocol (STP) is used to create a loop-free topology in Layer 2 networks.

STP is a Layer 2 protocol that calculates the best path through a switched network that contains redundant paths. STP uses bridge protocol data unit (BPDU) packets to exchange information with other switches. STP uses the information provided by the BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, and prune specific redundant links to create a loop-free tree topology. The resulting tree topology provides a single active Layer 2 data path between any two end stations. In discussions of STP, the terms *bridge* and *switch* are used interchangeably.

The original Spanning Tree Protocol is defined in the IEEE 802.1D 1998 specification. A newer version called Rapid Spanning Tree Protocol (RSTP) was originally defined in the IEEE 802.1w draft specification and later incorporated into the IEEE 802.1D-2004 specification. A recent version called Multiple Spanning Tree Protocol (MSTP) was originally defined in the IEEE 802.1s draft specification and later incorporated into the IEEE 802.1Q-2003 specification.

RSTP provides faster reconvergence time than the original STP by identifying certain links as point to point and by using protocol handshake messages rather than fixed timeouts. When a point-to-point link fails, the alternate link can transition to the forwarding state without waiting for any protocol timers to expire.

MSTP provides the capability to logically divide a Layer 2 network into regions. Every region has a unique identifier and can contain multiple instances of spanning trees. All regions are bound together using a Common Instance Spanning Tree (CIST), which is responsible for creating a loop-free topology across regions, whereas the Multiple Spanning Tree Instance (MSTI) controls topology inside regions. MSTP uses RSTP as a converging algorithm and is fully interoperable with earlier versions of STP.

The VLAN Spanning Tree Protocol (VSTP) is compatible with the Per-VLAN Spanning Tree Plus (PVST+) and Rapid-PVST+ protocols supported on Cisco Systems routers and switches. VSTP maintains a separate spanning-tree instance for each VLAN. Different VLANs can use different spanning-tree paths and VSTP can support up to 4094 different spanning-tree topologies. When different VLANs can use different spanning-tree paths, the CPU processing resources being consumed increase as more VLANs are configured. VSTP BPDU packets are tagged with the corresponding VLAN identifier and are transmitted to the multicast destination media access control (MAC) address 01-00-0c-cc-cc-cd with a protocol type of 0x010b. VSTP BPDUs are tunneled by pure IEEE 802.1q bridges.

The MX-series routers support STP, RSTP, MSTP, and VSTP.

NOTE: All virtual switch routing instances configured on an MX-series router are supported using only one spanning-tree process. The Layer 2 control protocol process is named l2cpd.

For more information about the various versions of STP, see the appropriate IEEE specification.

Configuring the Rapid Spanning Tree Protocol

This section discusses configuration statements and options for RSTP. Most of these statements also apply to MSTP and VSTP.

Enabling a Spanning-Tree Protocol

On an MX-series router you can enable the use of a spanning-tree protocol under a user-created routing instance of type virtual-switch or layer2-control. Configure the version of spanning-tree protocol to be used as RSTP, MSTP, or VSTP.

```
(rstp | mstp | vstp);
```

You can configure this statement at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *routing-instance-name* protocols]

Configuring the BPDU Destination MAC Address

A provider network can bridge the customer STP BPDU packets between customer sites by default. Simultaneously the provider network can prevent forwarding loops using STP in the provider network.

To configure a bridge to participate in the provider RSTP instance, include the following statement:

```
bpdu-destination-mac-address provider-bridge-group;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rstp]
- [edit routing-instances *routing-instance-name* protocols rstp]

When the provider-bridge-group option is specified, the destination MAC address of the BPDU packets transmitted is the provider bridge group address 01:80:c2:00:00:08, as defined in the IEEE 802.1ad specification. Received BPDU packets with this destination MAC address are accepted and passed to the Routing Engine.

Configuring the Bridge Priority

Use the bridge priority to control which bridge is elected as the root bridge. Also use the bridge priority to control which bridge is elected the root bridge when the initial root bridge fails.

The root bridge for each STP instance is determined by the bridge ID. The bridge ID consists of a configurable bridge priority and the MAC address of the bridge. The bridge with the lowest bridge ID is elected as the root bridge. If the bridge priorities are equal or if the bridge priority is not configured, the bridge with the lowest MAC address is elected the root bridge.

The bridge priority can also be used to determine which bridge becomes the designated bridge for a LAN segment. If two bridges have the same path cost to the root bridge, the bridge with the lowest bridge ID becomes the designated bridge.

The bridge priority can be set only in increments of 4096.

To configure the bridge priority, include the following statement:

```
bridge-priority priority;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp)]
- [edit protocols mstp msti *msti-id*]
- [edit protocols vstp vlan *vlan-id*]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp)]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Configuring the Maximum Age Timer

The maximum age timer specifies the maximum expected arrival time of hello BPDUs. If the maximum age timer expires, the bridge detects that the link to the root bridge has failed and initiates a topology reconvergence. The maximum age timer should be longer than the configured hello timer.

To configure the maximum age timer, include the following statement:

```
max-age seconds;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp)]
- [edit protocols vstp vlan *vlan-id*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Configuring the Hello Timer

The hello timer specifies the time interval at which the root bridge transmits configuration BPDUs.

To configure the hello timer, include the following statement:

```
hello-time seconds;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp)]
- [edit protocols vstp vlan *vlan-id*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Forcing the Spanning-Tree Version

The force-version statement forces the spanning-tree version to run as the original IEEE 802.1D version. Use this statement for compatibility with older bridges that do not support RSTP or VSTP.

To force the spanning-tree version, include the following statement:

```
force-version;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (rstp | vstp)]
- [edit routing-instances *routing-instance-name* protocols (rstp | vstp)]

Configuring the Forwarding Delay

The forwarding delay timer specifies the length of time an STP bridge port remains in the listening and learning states before transitioning to the forwarding state. Setting the interval too short could cause unnecessary spanning-tree reconvergence. Before changing this parameter, you should have a thorough understanding of STP.

To configure the forwarding delay timer, include the following statement:

```
forward-delay seconds;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp)]
- [edit protocols vstp vlan *vlan-id*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Configuring the Extended System Identifier

The extended system identifier is used to specify different bridge identifiers for different RSTP or STP routing instances.

To configure the extended system identifier, include the following statement:

```
extended-system-id identifier;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols rstp]
- [edit routing-instances *routing-instance-name* protocols rstp]

Configuring the Interface

STP and RSTP are limited to a single instance on any physical interface. Use the interface statement to configure which interfaces participate in the STP or RSTP instance. MSTP supports multiple instances on a single physical interface. Use the interface statement to configure which logical interfaces participate in MSTP.

For VSTP, interfaces can be configured at the global level or at the VLAN level. Interfaces configured at the global VSTP level will be enabled for all the configured VLANs. If an interface is configured at both the global and VLAN levels, the configuration at the VLAN level overrides the global configuration.

To configure the interface, include the following statements:

```
interface interface-name {
    cost cost;
    edge;
    mode (p2p | shared);
    priority interface-priority;
}
```

You can configure these statements at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)]
- [edit protocols vstp vlan *vlan-id*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Configuring the Interface Priority

The root port is the interface on the nonroot bridge with the lowest path cost to the root bridge. When multiple interfaces have the same path cost to the root bridge, the interface with the lowest interface priority is selected as the root port.

If the interface priority is not configured and multiple interfaces have the same path cost to the root bridge, the interface with the lowest interface identifier is selected as the root port.

If the interface priority is configured under the MSTP protocol, this becomes the default value for all interfaces. If the interface priority is configured under the MSTI interface, the value overrides the default for that interface.

If the interface priority is configured at both the VSTP global and VLAN levels, the configuration at the VLAN level overrides the global configuration.

To configure the interface priority, include the following statement:

priority *interface-priority*;

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit protocols mstp msti *msti-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*]
- [edit protocols vstp vlan *vlan-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Configuring the Interface Cost

The path cost used to calculate the root path cost from any given LAN segment is determined by the total cost of each link in the path. By default, the link cost is determined by the speed of the link. The interface cost can be configured to override the default cost and control which bridge is the designated bridge and which port is the designated port. In MSTP the CIST external path cost is determined by the link speed and the number of hops.

If the interface cost is not configured, the cost is determined by the speed of the interface. For example, a 100-Mbps link has a default path cost of 19, a 1000-Mbps link has a default path cost of 4, and a 10-Gbps link has a default path cost of 2.

If the interface cost is configured under MSTP, this becomes the default value for all interfaces. If the interface cost is configured under the MSTI interface, the value overrides the default for that interface.

If the interface cost is configured at both the VSTP global and VLAN levels, the configuration at the VLAN level overrides the global configuration.

The interface cost should be set the same for all interfaces connected to the same LAN segment.

To configure the interface cost, include the following statement:

cost *cost*;

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit protocols mstp msti *msti-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*]

- [edit protocols vstp vlan *vlan-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Configuring the Interface Mode

The interface mode allows RSTP, MSTP, and VSTP to converge faster than the original STP on point-to-point links. The protocol does not need to wait for timers on point-to-point links. Configure interfaces that have a point-to-point link to another Layer 2 bridge as p2p. This parameter is ignored if the STP is configured to run the original spanning-tree version.

If the interface mode is configured at both the VSTP global and VLAN levels, the configuration at the VLAN level overrides the global configuration.

To configure the interface mode, include the following statement:

```
mode (p2p | shared);
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit protocols vstp vlan *vlan-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Configuring an Edge Port

An edge port allows RSTP and MSTP to converge faster than the original STP. The protocol does not need to wait for BPDUs to be received on edge ports. Configure interfaces that are not connected to any Layer 2 bridge as edge ports. The JUNOS software supports automatic identification of edge ports as described in the RSTP standard. This parameter is ignored if the STP is configured to run the original spanning-tree version.

If the edge port is configured at both the VSTP global and VLAN levels, the configuration at the VLAN level overrides the global configuration.

To configure the interface as an edge port, include the following statement:

```
edge;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit protocols mstp msti *msti-id* interface *interface-name*]

- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*]
- [edit protocols vstp vlan *vlan-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Configuring Root Protect

Root protect helps to enforce the STP root bridge placement in a Layer 2 switched network. Enable root protect on interfaces that should not receive superior BPDUs from the root bridge. Typically, these ports are STP-designated ports on an administrative boundary.

If the bridge receives superior STP BPDUs on a port that has root protect enabled, that port is transitioned to a root-prevented STP state and the interface is blocked. This prevents a bridge that should not be the root bridge from being elected the root bridge.

After the bridge stops receiving superior STP BPDUs on the port with root protect enabled and the received BPDUs time out, that port is transitioned back to the STP designated port state.

When root protect is enabled on an interface, it is enabled for all STP instances on that interface. The interface is blocked only for those instances that receive superior BPDUs.

By default, root protect is disabled. To enable root protect, include the following statement:

```
no-root-port;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit protocols vstp vlan *vlan-id* interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*]
- [edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Tracing STP Traffic

To trace STP traffic, you can specify options in the global **traceoptions** statement included at the **[edit routing-options]** hierarchy level, and you can specify STP-specific options by including the **traceoptions** statement:

```
traceoptions {
```

```
    file filename <replace> <size size> <files number> <no-stamp> <world-readable |
      no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for the STP traceoptions statement.

You can specify the following STP-specific options in the STP traceoptions statement:

- all—Trace all operations.
- all-failures—Trace all failure conditions.
- bpdu—Trace BPDU reception and transmission.
- bridge-detection-state-machine—Trace the bridge detection state machine.
- events—Trace events of the protocol state machine.
- port-information-state-machine—Trace the port information state machine.
- port-migration-state-machine—Trace the port migration state machine.
- port-receive-state-machine—Trace the port receive state machine.
- port-role-transit-state-machine—Trace the port role transit state machine.
- port-role-select-state-machine—Trace the port role selection state machine.
- port-transmit-state-machine—Trace the port transmit state machine.
- port-state-transit-state-machine—Trace the port state transit state machine.
- ppmd—Trace the state and events for the ppmd process.
- state-machine-variables—Trace when the state machine variables change.
- timers—Trace protocol timers.
- topology-change-state-machine—Trace the topology change state machine.

NOTE: Use the trace flag all with caution. This flag may cause the CPU to become very busy.

For general information about tracing and global tracing options, see the statement summary for the global traceoptions statement in the *JUNOS Routing Protocols Configuration Guide*.

Example: Tracing STP Traffic

Trace only unusual or abnormal operations to /var/log/stp-log:

```
[edit]
routing-options {
   traceoptions {
      file /var/log/routing-log;
      flag errors;
```

```
    }
}
protocols {
    rstp {
        traceoptions {
            file /var/log/stp-log;
        }
    }
}
```

Configuring the Multiple Spanning Tree Protocol

The following sections discuss the parameters that are specific to MSTP:

Configuring the MSTP MSTI Instance Identifier

Each MSTP Multiple Spanning Tree Instance (MSTI) is identified by a number. The Common Instance Spanning Tree (CIST) is always MSTI ID 0. Each instance of an MSTI can be numbered 1 through 64. MSTI IDs are local to each region.

To configure the MSTI instance identifier, include the following statements:

```
msti msti-id {
    bridge-priority priority;
    vlan vlan-id;
    interface interface-name {
        cost cost;
        edge;
        priority interface-priority;
    }
}
```

You can configure these statements at the following hierarchy levels:

- [edit protocols mstp]
- [edit routing-instances *routing-instance-name* protocols mstp]

Configuring the MSTP Region Configuration Name

The configuration name is the MSTP region name carried in the MSTP BPDUs. The configuration name can be a maximum of 32 characters. The configuration name

helps define the logical boundary of the network. All switches in an MSTP region must have the same configuration name configured.

To configure the configuration name, include the following statement:

```
configuration-name configuration-name;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp]
- [edit routing-instances *routing-instance-name* protocols mstp]

Configuring the MSTP Revision Level

The MSTP revision level is the revision number of the configuration. All switches in an MSTP region must have the same revision level configured.

To configure the MSTP revision level, include the following statement:

```
revision-level revision-level;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp]
- [edit routing-instances *routing-instance-name* protocols mstp]

Configuring the MSTP Maximum Hops

The MSTP maximum hops value is the maximum number of hops in the region. The MSTI root bridge sends BPDUs with the hop count set to the maximum value. When a bridge receives this BPDU, it decrements the remaining hop count by one and propagates this hop count in the BPDUs it sends. When a bridge receives a BPDU with a hop count of zero, the bridge discards the BPDU.

To configure the MSTP maximum hops, include the following statement:

```
max-hops hops;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp]
- [edit routing-instances *routing-instance-name* protocols mstp]

Configuring the MSTI Interface

To configure the MSTI logical interface-specific parameters, include the following statement:

```
interface interface;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp msti *msti-id*]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*]

Configuring the MSTI VLAN

An MSTI can map to a range of VLANs just as a logical port can map to a range of VLANs. The MSTP VLAN specifies the VLAN or VLAN range to which this MSTI is mapped. The vlan-id is configured under the logical interface.

To configure the VLAN, include the following statement:

```
vlan vlan-id;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp msti *msti-id*]
- [edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*]

Disabling the MSTP Instance

To disable the entire MSTP instance, include the following statement:

```
disable;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mstp]
- [edit routing-instances *routing-instance-name* protocols mstp]

Configuring the VLAN Spanning Tree Protocol

This section describes configuration statements for the VLAN Spanning Tree Protocol (VSTP). For VSTP, the bridge-priority, max-age, hello-time, forward-delay, priority, cost, mode, and edge statements all have the same meaning as they do for the standard STP, RSTP, and MSTP values.

The following sections discuss the parameters that are specific to VSTP:

- VSTP Limitations on page 99
- Configuring a VSTP VLAN Instance on page 100

VSTP Limitations

VSTP cannot be configured on a virtual switch if any of the virtual switch bridge domains contain ports with VLAN ranges or VLAN mappings.

To enable VSTP for a specific VLAN ID, there must be a bridge domain or VPLS routing instance with the same VLAN ID and all the logical interfaces assigned to the VLAN must have the same matching VLAN ID.

Configuring a VSTP VLAN Instance

To enable a VSTP instance for a specified VLAN, include the vlan statement:

```
vlan vlan-id;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols vstp]
- [edit routing-instances *routing-instance-name* protocols vstp]

Configuring Layer 2 Protocol Tunneling

Layer 2 protocol tunneling allows Layer 2 protocol data units (PDUs) to be tunneled through a network. This is useful to provide a single STP domain for subscribers across a service provider network. It is also useful for tunneling Cisco Discovery Protocol (CDP) or VLAN Trunk Protocol (VTP) PDUs across a network.

When a control packet for STP, CDP, or VTP is received on a service provider edge port configured for Layer 2 protocol tunneling, the multicast destination MAC address is rewritten with the predefined multicast tunnel MAC address of 01:00:0c:cd:cd:d0. The packet is transported across the provider network transparently to the other end of the tunnel and the original multicast destination MAC address is restored when the packet is transmitted.

If a packet is received on a tunnel interface that already has a destination multicast MAC address of 01:00:0c:cd:cd:d0, the port enters an error state and is shut down. To clear the error condition, the administrator must enter the clear error mac-rewrite interface *interface-name* command.

Layer 2 protocol tunneling is supported on MX-series routers with enhanced queueing Dense Port Concentrators (DPCs).

Enabling Layer 2 Protocol Tunneling

To enable the Layer 2 protocol tunneling feature, include the mac-rewrite statement at the [edit protocols layer2-control] hierarchy level:

```
[edit protocols layer2-control]
mac-rewrite;
```

Configure the mac-rewrite statement only on untagged and single identifier tagged interfaces, and not on double identifier tagged interfaces. For tagged ports, configure

a logical interface with the native VLAN identifier. This configuration associates the untagged control packets with a logical interface.

The destination multicast tunnel MAC address of 01:00:0c:cd:cd:d0 is installed in the MAC table when the mac-rewrite statement is configured.

Configuring the Layer 2 Protocol Tunnel Interface

The Layer 2 protocol tunneling configuration must be done on the interfaces at each end of the tunnel.

To configure the interface where Layer 2 protocol tunneling is enabled, include the interface ge-*fpc/pic/port* statement at the [edit protocols layer2-control mac-rewrite] hierarchy level:

```
[edit protocols layer2-control mac-rewrite]
interface ge-fpc/pic/port;
```

Configuring the Layer 2 Protocol to be Tunneled

To configure the protocol that is tunneled by the Layer 2 protocol tunnel, include the protocol (cdp | stp | vtp) statement at the [edit protocols layer2-control mac-rewrite interface ge-*fpc/pic/port*] hierarchy level:

```
[edit protocols layer2-control mac-rewrite interface ge-fpc/pic/port]
protocol (cdp | stp | vtp);
```

For each protocol specified, a static destination MAC address corresponding to the protocol being tunneled is installed in the MAC table.

When CDP, STP, or VTP is configured for tunneling on a customer-facing port in a provider bridge, the corresponding protocol should not be enabled for operation on that interface.

Configuring Layer 2 Control BPDU Protection

The Spanning Tree Protocol (STP) family is designed to break possible loops in a Layer 2 bridged network. Loop prevention avoids damaging broadcast storms that can potentially render the network useless. STP processes on bridges exchange bridge protocol data units (BPDUs) to determine the LAN topology, decide the root bridge, stop forwarding on some ports, and so on. However, a misbehaving user application or device can interfere with the operation of the STPs and cause network problems.

On the MX-series routers only, you can configure BPDU protection to ignore BPDUs received on interfaces where none should be expected (for example, a LAN interface on a network edge with no other bridges present). If a BPDU is received on a blocked interface, the interface is disabled and stops forwarding frames. By default, all BPDUs are accepted and processed on all interfaces.

You can enable BPDU protection on individual interfaces or on all the edge ports of the bridge.

You can configure BPDU protection on interfaces with the following encapsulation types:

- ethernet-bridge
- etherent-vpls
- extended-vlan-bridge
- vlan-vpls
- extended-vlan-vpls

To configure BPDU blocking on one or more interfaces, include the bpdu-block statement:

```
[edit protocols layer2-control]
bpdu-block {
    interface interface-name;
    disable-timeout seconds;
}
```

You can configure these statements at the following hierarchy levels:

- [edit protocols layer2-control]
- [edit routing-instances *routing-instance-name* protocols layer2–control]

To configure the interfaces on which the system should not expect to receive BPDUs, include the interfaces *interface-name* statement at the [edit protocols layer2-control bpdu-block] hierarchy level. You can apply this statement to aggregated Ethernet interfaces. By default, the system accepts all BPDUs received on any interface unless you include this statement. If you configure this statement on a blocked interface, and a BPDU is received on the interface, the system will disable the interface and stop forwarding frames out the interface until the bridging process is restarted. You can alter this behavior with the disable-timeout statement.

To configure the amount of time that interfaces should wait before enabling a blocked interface that has received a BPDU, include the disable-timeout *seconds* statement at the [edit protocols layer2-control bpdu-block] hierarchy level. By default, if a BPDU is received on a blocked interface, the system will disable the interface and stop forwarding frames out the interface until the interface is cleared. You can alter this behavior with the disable-timeout statement. You specify the time the system waits before unblocking the interface that has received the BPDU. The range is from 10 through 3600 seconds (one hour). A disable-timeout value of 0 is allowed, but this results in the default behavior (the interface is blocked until the interface is cleared).

The following example, when used with a full bridge configuration with aggregated Ethernet, blocks BPDUs on aggregated interface ae0 for ten minutes (600 seconds) before enabling the interface again:

```
[edit protocols layer2-control]
bpdu-block {
    interface ae0;
    disable-timeout 600;
}
```

You check the status of the interface with the show interfaces command. If the value of the BPDU Error field is Detected and the link is down, the interface is blocked. If the interface is enabled, the value of the BPDU Error field should be none.

You clear the blocked status of an interface with the clear error bpdu interface *interface-name* command. (Note that the disable-timeout interval will automatically clear interfaces after the specified interval unless the interval is 0.)

In some cases, the topology determined by one STP bridge protocol might differ from the topology determined by another STP family member. In this case, edge ports to MSTP (for example) might not be edge ports to VSTP. You can block a particular STP family member by blocking BPDU reception on edge ports that should not be receiving BPDUs. In contrast to the bpdu-block statement, bpdu-block-on-edge disables designated edge ports and does not enable them again.

To configure edge port blocking for a particular STP family member, include the bpdu-block-on-edge statement for mstp, rstp, or vstp:

```
[edit]
protocols {
   ( mstp | rstp | vstp ) {
      bpdu-block-on-edge;
      interface interface-name;
   }
}
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)]

You must still fully configure the interfaces and STP protocol.

Configuring STP Loop Protection

The Spanning Tree Protocol (STP) family is responsible for breaking loops in a network of bridges with redundant links. However, hardware failures can create forwarding loops (STP loops) and cause major network outages. STP breaks loops by blocking ports (interfaces). Errors occur when a blocked port transitions erroneously to a forwarding state.

Ideally, an STP port remains blocked as long as a superior alternate path to the root bridge exists for a connected LAN segment. This designated port is determined by receiving superior bridge protocol data units (BPDUs) from a peer on that port. When other ports no longer receive BPDUs, the STP considers the topology to be loop free. However, if a blocked or alternate port moves into a forwarding state, this creates a loop.

You can configure STP loop protection to improve the stability of Layer 2 networks. STP loop protection enhances the normal checks the STP performs on interfaces by performing a specified action when BPDUs are not received on a non-designated port interface. You can choose to block the interface or issue an alarm when BPDUs are not received on the port. By default (that is, without STP loop protection

configured), an interface that stops receiving BPDUs will assume the designated port role and possibly result in an STP loop. You configure STP loop protection to prevent selected interfaces from interpreting the lack of BPDUs as a "false positive" for making the interface the designated port. STP loop protection is enabled for all STP instances on the interface, but blocks or alarms only those instances that stop receiving BPDUs.

To configure STP loop protection, include the **bpdu-timeout-action** statement with either the **block** or **alarm** option for the STP interface:

```
[edit protocols]
mstp {
      interface interface-name {
         bpdu-timeout-action ( block | alarm );
      }
}

rstp {
      interface interface-name {
         bpdu-timeout-action ( block | alarm );
      }
}

vstp {
      interface interface-name {
         bpdu-timeout-action ( block | alarm );
      }
      vlan vlan-id {
         interface interface-name {
            bpdu-timeout-action ( block | alarm );
         }
      }
}
```

You can configure this statement at the following hierarchy levels:

- [edit protocols (mstp | rstp | vstp)]
- [edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)]

This example blocks the non-designated RSTP port **ge-1/2/0** after the BPDU timeout interval expires:

```
[edit protocols]
rstp {
   interface ge-1/2/0 {
      bpdu-timeout-action block;
   }
}
```

You must still fully configure the interfaces and RSTP protocol.

You can display the loop protection characteristics on an interface using the **show spanning-tree interface** command.

Chapter 9

Summary of Spanning Tree Protocol Configuration Statements

The following sections explain each of the Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP) configuration statements. The statements are organized alphabetically.

bpdu-block

Syntax

```
bpdu-block {
    interface interface-name;
    disable-timeout seconds;
}
```

Hierarchy Level [edit protocols layer2-control],
[edit routing-instances *routing-instance-name* protocols layer2-control]

Release Information Statement introduced in JUNOS Release 9.4.

Description Enable BPDU blocking on an interface.

The remaining statements are described separately.

Usage Guidelines See "Configuring Layer 2 Control BPDU Protection" on page 101.

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

bpdu-block-on-edge

Syntax bpdu-block-on-edge;

Hierarchy Level [edit protocols mstp],
[edit protocols rstp],
[edit protocols vstp],
[edit routing-instances *routing-instance-name* protocols mstp],
[edit routing-instances *routing-instance-name* protocols rstp],
[edit routing-instances *routing-instance-name* protocols vstp]

Release Information Statement introduced in JUNOS Release 9.4.

Description Enable BPDU blocking on the edge ports of a virtual switch.

Usage Guidelines See "Configuring Layer 2 Control BPDU Protection" on page 101.

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

bpdu-destination-mac-address

Syntax bpdu-destination-mac-address provider-bridge-group;

Hierarchy Level [edit protocols rstp],
[edit routing-instances *routing-instance-name* protocols rstp]

Release Information Statement introduced in JUNOS Release 9.2.

Description Participate in the provider Rapid Spanning Tree Protocol (RSTP) instance.

Default If the bpdu-destination-mac-address statement is not configured, the bridge participates in the customer RSTP instance, transmitting and receiving standard RSTP BPDU packets.

Options provider-bridge-group—The destination MAC address of the BPDU packets transmitted is the provider bridge group address 01:80:c2:00:00:08. Received BPDU packets with this destination MAC address are accepted and passed to the Routing Engine.

Usage Guidelines See "Configuring the BPDU Destination MAC Address" on page 89

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

bpdu-timeout-action

Syntax bpdu-timeout-action (block | alarm);

Hierarchy Level [edit protocols (mstp | rstp | vstp}],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)]

Release Information Statement introduced in JUNOS Release 9.4.

Description Provide STP loop protection for a given STP family protocol interface.

Default If the bpdu-timeout-action statement is not configured, an interface that stops receiving BPDUs will transition to the designated port (forwarding) state, creating a potential loop.

Options block—The interface is blocked if it has not received BPDUs during the timeout interval.

alarm—The interface raises an alarm condition if it has not received BPDUs during the timeout interval.

Usage Guidelines See "Configuring STP Loop Protection" on page 103.

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

bridge-priority

Syntax bridge-priority *priority*;

Hierarchy Level [edit protocols (mstp | rstp)],
[edit protocols mstp msti *msti-id*],
[edit protocols vstp vlan *vlan-id*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Determine which bridge is elected as the root bridge. If two bridges have the same path cost to the root bridge, the bridge priority determines which bridge becomes the designated bridge for a LAN segment.

Options *priority*—The bridge priority can be set only in increments of 4096.
Range: 0 through 61,440
Default: 32,768

Usage Guidelines See "Configuring the Bridge Priority" on page 89.

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

configuration-name

Syntax configuration-name *configuration-name*;

Hierarchy Level [edit protocols mstp],
[edit routing-instances *routing-instance-name* protocols mstp]

Release Information Statement introduced in JUNOS Release 8.4.

Description The configuration name is the MSTP region name carried in the MSTP BPDUs.

Usage Guidelines See "Configuring the MSTP Region Configuration Name" on page 97.

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

cost

Syntax cost *cost*;

Hierarchy Level [edit protocols (mstp | rstp | vstp) interface *interface-name*],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols vstp vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure link cost to control which bridge is the designated bridge and which port is the designated port. By default, the link cost is determined by the link speed.

Options *cost*—(Optional) Link cost associated with the port.
Range: 1 through 200,000,000

Usage Guidelines See "Configuring the Interface Cost" on page 93.

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

disable

Syntax disable;

Hierarchy Level [edit protocols mstp],
[edit routing-instances *routing-instance-name* protocols mstp]

Release Information Statement introduced in JUNOS Release 9.1.

Description Disable the entire MSTP instance.

Usage Guidelines See "Disabling the MSTP Instance" on page 99

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

disable-timeout

Syntax disable-timeout *seconds*;

Hierarchy Level [edit protocols layer2-control bpdu-block],
[edit routing-instances *routing-instance-name* protocols layer2-control bpdu-block]

Release Information Statement introduced in JUNOS Release 9.4.

Description Configure the timeout value to periodically check to see if an interface is still disabled with BPDU blocking. If this option is not configured, the interface is not periodically checked and remains disabled.

Options *seconds*—Disable timeout value.
Range: 10 through 3600
Default: If this option is not configured, the interface is not periodically checked and remains disabled.

Usage Guidelines See "Configuring Layer 2 Control BPDU Protection" on page 101.

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

edge

Syntax edge;

Hierarchy Level [edit protocols (mstp | rstp | vstp) interface *interface-name*],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols vstp vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure interfaces as edge ports. Edge ports do not expect to receive BPDUs. If a BPDU is received, the port becomes a nonedge port.

Usage Guidelines See "Configuring an Edge Port" on page 94.

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

extended-system-id

Syntax extended-system-id <*identifier*>;

Hierarchy Level [edit protocols rstp],
[edit routing-instances *routing-instance-name* protocols rstp]

Release Information Statement introduced in JUNOS Release 8.3.

Description The extended system ID is used to specify different bridge identifiers for different RSTP or STP routing instances.

Options *identifier*—Specify the system identifier to use for the RSTP or STP instance.
Range: 0 through 4095

Usage Guidelines See "Configuring the Extended System Identifier" on page 91

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

force-version

Syntax force-version;

Hierarchy Level [edit protocols (rstp | vstp)],
[edit routing-instances *routing-instance-name* protocols (rstp | vstp)]

Release Information Statement introduced in JUNOS Release 8.4.

Description Force the spanning-tree version to be the original IEEE 803.1D STP.

Usage Guidelines See "Forcing the Spanning-Tree Version" on page 91.

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

forward-delay

Syntax forward-delay *seconds*;

Hierarchy Level [edit protocols (mstp | rstp)],
[edit protocols vstp vlan *vlan-id*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Specify the length of time an STP bridge port remains in the listening and learning states before transitioning to the forwarding state.

Options *seconds*—(Optional) Number of seconds the bridge port remains in the listening and learning states.
Range: 4 through 30
Default: 15 seconds

Usage Guidelines See "Configuring the Forwarding Delay" on page 91.

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

hello-time

Syntax hello-time *seconds*;

Hierarchy Level [edit protocols (mstp | rstp)],
[edit protocols vstp vlan *vlan-id*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Specify the number of seconds between transmissions of configuration BPDUs by the root bridge.

Options *seconds*—(Optional) Number of seconds between transmissions of configuration BPDUs.
Range: 1 through 10
Default: 2 seconds

Usage Guidelines See "Configuring the Hello Timer" on page 90.

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

interface

See the following sections:

interface (Layer 2 Protocol Tunneling)

Syntax

```
interface interface-name {
    protocol (cdp | stp | vtp);
}
```

Hierarchy Level [edit protocols layer2-control mac-rewrite],
[edit routing-instances *routing-instance-name* protocols layer2-control mac-rewrite]

Release Information Statement introduced in JUNOS Release 9.1.

Description Configure an interface for Layer 2 protocol tunneling.

The remaining statements are described separately.

Usage Guidelines See "Configuring the Layer 2 Protocol Tunnel Interface" on page 101.

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

interface (Spanning Tree)

Syntax
```
interface interface-name {
    cost cost;
    edge;
    mode (p2p | shared);
    no-root-port;
    priority interface-priority;
}
```

Hierarchy Level
[edit protocols (mstp | rstp | vstp)],
[edit protocols vstp vlan *vlan-id*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure the interface to participate in the RSTP or MSTP instance.

Options *interface-name*—Name of a Gigabit Ethernet or 10-Gigabit Ethernet interface.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Interface" on page 92.

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

interface (BPDU Blocking)

Syntax
```
interface interface-name;
```

Hierarchy Level
[edit protocols layer2-control bpdu-block],
[edit routing-instances *routing-instance-name* protocols layer2-control bpdu-block]

Release Information Statement introduced in JUNOS Release 9.4.

Description Configure the interface to participate BPDU blocking.

Options *interface-name*—Name of a Gigabit Ethernet or 10-Gigabit Ethernet interface.

Usage Guidelines See "Configuring Layer 2 Control BPDU Protection" on page 101.

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

layer2-control

Syntax

```
layer2-control {
    bpdu-block {
        interface interface-name;
        disable-timeout seconds;
    }
    mac-rewrite {
        interface interface-name {
            protocol (cdp | stp | vtp);
        }
    }
    nonstop-bridging;
}
```

Hierarchy Level [edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced in JUNOS Release 8.4.
bpdu-block option added in JUNOS Release 9.4.

Description Configure Layer 2 control protocols to enable features such as Layer 2 protocol tunneling or nonstop bridging.

The remaining statements are described separately.

Usage Guidelines See "Configuring Layer 2 Protocol Tunneling" on page 100 and "Configuring Layer 2 Control BPDU Protection" on page 101.

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

Related Topics nonstop-bridging statement in the *JUNOS High Availability Configuration Guide*.

mac-rewrite

Syntax
```
mac-rewrite {
    interface interface-name {
        protocol (cdp | stp | vtp);
    }
}
```

Hierarchy Level [edit protocols layer2-control],
[edit routing-instances *routing-instance-name* protocols layer2-control]

Release Information Statement introduced in JUNOS Release 9.1.

Description Enable rewriting of the MAC address for Layer 2 protocol tunneling.

The remaining statements are described separately.

Usage Guidelines See "Enabling Layer 2 Protocol Tunneling" on page 100

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

max-age

Syntax max-age *seconds*;

Hierarchy Level [edit protocols (mstp | rstp)],
[edit protocols vstp vlan *vlan-id*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Specify the maximum expected arrival time of hello BPDUs.

Options *seconds*—(Optional) Number of seconds expected between hello BPDUs.
Range: 6 through 40
Default: 20 seconds

Usage Guidelines See "Configuring the Maximum Age Timer" on page 90.

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

max-hops

Syntax max-hops *hops*;

Hierarchy Level [edit protocols mstp],
[edit routing-instances *routing-instance-name* protocols mstp]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure the maximum number of hops a BPDU can be forwarded in the MSTP region.

Options *hops*—(Optional) Number of hops the BPDU can be forwarded.
Range: 1 through 255
Default: 19 hops

Usage Guidelines See "Configuring the MSTP Maximum Hops" on page 98.

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

mode

Syntax mode (p2p | shared);

Hierarchy Level [edit protocols (mstp | rstp | vstp) interface *interface-name*],
[edit protocols vstp vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure link mode to identify point-to-point links.

Default When the link is configured as full-duplex, the default link mode is p2p. When the link is configured half-duplex, the default link mode is shared.

Options p2p—The link is point to point.

shared—The link is shared media.

Usage Guidelines See "Configuring the Interface Mode" on page 94.

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

msti

Syntax

```
msti msti-id {
    bridge-priority priority;
    vlan vlan-id;
    interface interface-name {
        cost cost;
        edge;
        priority interface-priority;
    }
}
```

Hierarchy Level [edit protocols mstp],
[edit routing-instances *routing-instance-name* protocols mstp]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure the Multiple Spanning Tree Protocol (MSTI) instance identifier.

Options *msti-id*—MSTI instance identifier.
Range: 1 through 64

The remaining statements are explained separately.

Usage Guidelines See “Configuring the MSTP MSTI Instance Identifier” on page 97.

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

mstp

Syntax

```
mstp {
    bpdu-block-on-edge;
    bridge-priority priority;
    configuration-name configuration-name;
    revision-level revision-level;
    max-hops hops;
    bridge-priority priority;
    max-age seconds;
    hello-time seconds;
    forward-delay seconds;
    interface interface-name {
        bpdu-timeout-action ( block | alarm );
        cost cost;
        edge;
        mode (p2p | shared);
        no-root-port;
        priority interface-priority;
    }
    msti msti-id {
        bridge-priority priority;
        vlan vlan-id;
        interface interface-name {
            cost cost;
            edge;
            priority interface-priority;
        }
    }
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp> <world-readable |
            no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level

[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in JUNOS Release 8.4.
bpdu-block-on-edge option added in JUNOS Release 9.4.
bpdu-timeout-action added in JUNOS Release 9.4.

Description

Configure MSTP parameters.

Options

The statements are explained separately.

Usage Guidelines

See "Configuring the Multiple Spanning Tree Protocol" on page 97 and "Configuring Layer 2 Control BPDU Protection" on page 101.

Required Privilege Level

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

no-root-port

Syntax no-root-port;

Hierarchy Level [edit protocols (mstp | rstp | vstp) interface *interface-name*],
[edit protocols vstp vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 9.1.

Description Ensure the port is the spanning-tree designated port. If the port receives superior bridge protocol data unit (BPDU) packets, root protect moves this port to a root-prevented spanning-tree state.

Usage Guidelines See "Configuring Root Protect" on page 95.

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

priority

Syntax priority *interface-priority*;

Hierarchy Level [edit protocols (mstp | rstp | vstp) interface *interface-name*],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols vstp vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp) interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Use the interface priority to control which interface is elected as the root port. The interface priority must be set in increments of 16.

Options *priority*—(Optional) Interface priority.
Range: 0 through 240

Usage Guidelines See "Configuring the Interface Priority" on page 92.

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

protocol

Syntax protocol (cdp | stp | vtp);

Hierarchy Level [edit protocols layer2-control mac-rewrite interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols layer2-control mac-rewrite interface *interface-name*]

Release Information Statement introduced in JUNOS Release 9.1.

Description Configure the protocol to be tunneled on an interface for Layer 2 protocol tunneling. To tunnel multiple protocols, include multiple protocol statements.

Options cdp—Tunnel the Cisco discovery protocol.

stp—Tunnel all versions of the spanning-tree protocol.

vtp—Tunnel the VLAN trunk protocol.

Usage Guidelines See "Configuring the Layer 2 Protocol to be Tunneled" on page 101.

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

protocols

Syntax protocols (mstp | rstp | vstp);

Hierarchy Level [edit routing-instances *routing-instance-name*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure the Spanning Tree Protocol type as RSTP or MSTP.

Options mstp—Configure the protocol as Multiple Spanning Tree.

rstp—Configure the protocol as Rapid Spanning Tree.

vstp—Configure the protocol as VLAN Spanning Tree.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Rapid Spanning Tree Protocol" on page 88, "Configuring the Multiple Spanning Tree Protocol" on page 97, and "Configuring the VLAN Spanning Tree Protocol" on page 99

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

revision-level

Syntax revision-level *revision-level*;

Hierarchy Level [edit protocols mstp],
[edit routing-instances *routing-instance-name* protocols mstp]

Release Information Statement introduced in JUNOS Release 8.4.

Description Set the revision number of the MSTP configuration.

Options *revision-level*—Configure the revision number of the MSTP region configuration.
Range: 0 through 65,535

Usage Guidelines See "Configuring the MSTP Revision Level" on page 98.

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

rstp

Syntax

```
rstp {
    bpdu-block-on-edge;
    bridge-priority priority;
    bpdu-destination-mac-address provider-bridge-group;
    max-age seconds;
    hello-time seconds;
    extended-system-id;
    force-version;
    forward-delay seconds;
    interface interface-name {
        bpdu-timeout-action ( block | alarm );
        cost cost;
        edge;
        mode (p2p | shared);
        no-root-port;
        priority interface-priority;
    }
    traceoptions {
        file filename <files number> <no-stamp> <replace> <size size> <world-readable |
            no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level

[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in JUNOS Release 8.4.
bpdu-block-on-edge option added in JUNOS Release 9.4.
bpdu-timeout-action option added in JUNOS Release 9.4.

Description

Configure RSTP parameters.

Options

The statements are explained separately.

Usage Guidelines

See “Configuring the Rapid Spanning Tree Protocol” on page 88 and “Configuring Layer 2 Control BPDU Protection” on page 101.

Required Privilege Level

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

traceoptions

Syntax

```
traceoptions {
   file name <replace> <size size> <files number> <no-stamp> <world-readable |
      no-world-readable>;
   flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit protocols (mstp | rstp | vstp)],
[edit routing-instances *routing-instance-name* protocols (mstp | rstp | vstp)]

Release Information

Statement introduced in JUNOS Release 8.4.

Description

Set STP protocol-level tracing options.

Default

The default STP protocol-level trace options are inherited from the global **traceoptions** statement.

Options

disable—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *name*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place STP tracing output in the file **/var/log/stp-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file*.0, then *trace-file*.1, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.
Range: 2 through 1000 files
Default: 1 trace file only

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The following are the STP-specific tracing options:

- **all**—Trace all operations.
- **all-failures**—Trace all failure conditions.
- **bpdu**—Trace BPDU reception and transmission.
- **bridge-detection-state-machine**—Trace the bridge detection state machine.
- **events**—Trace events of the protocol state machine.
- **port-information-state-machine**—Trace the port information state machine.

- port-migration-state-machine—Trace the port migration state machine.
- port-receive-state-machine—Trace the port receive state machine.
- port-role-transit-state-machine—Trace the port role transit state machine.
- port-role-select-state-machine—Trace the port role selection state machine.
- port-state-transit-state-machine—Trace the port state transit state machine.
- port-transmit-state-machine—Trace the port transmit state machine.
- ppmd—Trace the state and events for the ppmd process.
- state-machine-variables—Trace when the state machine variables change.
- timers—Trace protocol timers.
- topology-change-state-machine—Trace the topology change state machine.

The following are the global tracing options:

- all—All tracing operations.
- config-internal—Trace configuration internals.
- general—Trace general events.
- normal—All normal events.

  Default: If you do not specify this option, only unusual or abnormal operations are traced.
- parse—Trace configuration parsing.
- policy—Trace policy operations and actions.
- regex-parse—Trace regular-expression parsing.
- route—Trace routing table changes.
- state—Trace state transitions.
- task—Trace protocol task processing.
- timer—Trace protocol task timer processing.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.
Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.
Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When the *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the files option.
Syntax: *x*k to specify KB, *x*m to specify MB, or *x*g to specify GB
Range: 10 KB through the maximum file size supported on your system
Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See "Tracing STP Traffic" on page 95.

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

vlan

See the following sections:

vlan (MSTP)

Syntax vlan *vlan-id*;

Hierarchy Level [edit protocols mstp msti *msti-id*],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*]

Release Information Statement introduced in JUNOS Release 8.4.

Description Configure the VLAN of an MSTI or VSTP instance or configure the VLAN range of an MSTI instance.

Options *vlan-id*—The VLAN identifier associated with the MSTI.

min-vlan-id-max-vlan-id—Range of VLAN identifiers associated with the MSTI in the form *minimum-vlan-id–maximium-vlan-id*. VLAN identifier ranges are not supported for VSTP.
Range: 1 through 4096

Usage Guidelines See "Configuring the MSTI VLAN" on page 99.

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

vlan (VSTP)

Syntax

```
vlan vlan-id {
    bridge-priority priority;
    max-age seconds;
    hello-time seconds;
    forward-delay seconds;
    interface interface-name {
        cost cost;
        edge;
        mode (p2p | shared);
        no-root-port;
        priority interface-priority;
    }
}
```

Hierarchy Level [edit protocols vstp]

Release Information Statement introduced in JUNOS Release 9.0.

Description Configure VSTP VLAN parameters.

Options The statements are explained separately.

Usage Guidelines See “Configuring the VLAN Spanning Tree Protocol” on page 99.

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

vstp

Syntax

```
vstp {
    bpdu-block-on-edge;
    force-version (stp | rstp);
    interface interface-name {
        bpdu-timeout-action ( block | alarm );
        cost cost;
        edge;
        mode (p2p | shared);
        no-root-port;
        priority interface-priority;
    }
    vlan vlan-id {
        bridge-priority priority;
        max-age seconds;
        hello-time seconds;
        forward-delay seconds;
        interface interface-name {
            bpdu-timeout-action ( block | alarm );
            cost cost;
            edge;
            mode (p2p | shared);
            no-root-port;
            priority interface-priority;
        }
    }
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp> <world-readable |
            no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level

[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in JUNOS Release 9.0.
bpdu-block-on-edge option added in JUNOS Release 9.4.
bpdu-timeout-action added in JUNOS Release 9.4.

Description

Configure VSTP parameters.

Options

The statements are explained separately.

Usage Guidelines

See “Configuring the VLAN Spanning Tree Protocol” on page 99, “Configuring the Rapid Spanning Tree Protocol” on page 88, and “Configuring Layer 2 Control BPDU Protection” on page 101.

Required Privilege Level

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

Part 6
Indexes

Index

F

G

H

I

L

M

N

P

R

S

T

V

Index of Statements and Commands

S

T

V