JUNOS™ Software

MX-series Solutions Guide

Release 8.5

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

Part Number: 530-0211963-01, Revision 1

JUNOS™ Software MX-series Solutions Guide
Release 8.5

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

Revision History
12 October 2007—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 Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and 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, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper 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 the 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 software on multiple computers requires multiple licenses, regardless of whether such computers 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 the Embedded Software on non-Juniper equipment; (j) use the 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 for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

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

List of Figures

List of Tables

About This Guide

This preface provides the following guidelines for using the *JUNOS™ Software MX-series Solutions Guide*:

Objectives

This guide provides an overview of the Layer 2 features of the MX-series routers and describes how to configure the features to provide solutions to several network scenarios.

NOTE: This guide documents Release 8.5 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 MX-series routing platform.

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 xvi defines notice icons used in this guide.

Table 1: Notice Icons

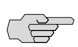
Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.

Table 2 on page xvi 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 xviii lists the software and hardware guides and release notes for Juniper Networks J-series, M-series, MX-series, and T-series routing platforms and describes the contents of each document. Table 4 on page xxi lists the books included in the *Network Operations Guide* series.

Table 5 on page xxii lists additional books on Juniper Networks solutions that you can order through your bookstore. A complete list of such books is available at 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.
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 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.
Multiplay Solutions	Describes how you can deploy IPTV and voice over IP (VoIP) services in your network.
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.

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

Book	Description
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.
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing, 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*.
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.

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

Book	Description
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.
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.
J-series Routing Platform Documentation	
Getting Started Guide	Provides an overview, basic instructions, and specifications for J-series routing platforms. The guide explains how to prepare your site for installation, unpack and install the router and its components, install licenses, and establish basic connectivity. Use the Getting Started Guide for your router model.

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

Book	Description
Basic LAN and WAN Access Configuration Guide	Explains how to configure the interfaces on J-series Services Routers for basic IP routing with standard routing protocols, ISDN backup, and digital subscriber line (DSL) connections.
Advanced WAN Access Configuration Guide	Explains how to configure J-series Services Routers in virtual private networks (VPNs) and multicast networks, configure data link switching (DLSw) services, and apply routing techniques such as policies, stateless and stateful firewall filters, IP Security (IPSec) tunnels, and class-of-service (CoS) classification for safer, more efficient routing.
Administration Guide	Shows how to manage users and operations, monitor network performance, upgrade software, and diagnose common problems on J-series Services Routers.
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.
J-series Services Router Release Notes	Briefly describe Services Router features, identify known hardware problems, and provide upgrade and downgrade instructions.

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.

Table 4: JUNOS Software Network Operations Guides *(continued)*

Book	Description
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.

Table 5: 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 http://www.juniper.net/techpubs/docbug/docbugreport.html. 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

Requesting Support

For technical support, open a support case with the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1
Overview

Chapter 1

Overview of Ethernet Solutions

The Ethernet LAN environment is very different than the IP routing environment. This overview chapter outlines many of those differences and introduces the terms, acronyms, and concepts used in Ethernet networking.

This chapter provides the following information about Ethernet networking solutions:

Terminology and Acronyms

Networking with a switch over Ethernet on a LAN is different than networking with a router with IP over a wider area. Even the words used to talk about Ethernet networking are different from those used in IP routing. This section provides a list of all the terms and acronyms used in this manual, as well terms that apply to a complete network using Ethernet as a carrier technology.

- 802.1ad—The IEEE specification for “Q-in-Q” encapsulation and bridging of Ethernet frames.
- 802.1ah—The IEEE specification for media access control (MAC) tunneling encapsulation and bridging of Ethernet frames across a provided backbone-managed bridge.
- 802.3ah—The IEEE specification for link fault management (LFM), a method for Operations, Administration, and Maintenance (OAM) of Ethernet links.
- 802.1Q—The IEEE specification for adding virtual local area network (VLAN) tags to an Ethernet frame.
- B–MAC—The backbone source and destination MAC address fields found in the IEEE 802.1ah provider MAC encapsulation header.
- Bridge—A network component defined by the IEEE that forwards frames from one LAN segment or VLAN to another. The bridging function can be contained in a router, LAN switch, or other specialized device. See also *switch*.
- Bridge domain—A set of logical ports that share the same flooding or broadcast characteristics. As in a virtual LAN, a bridge domain spans one or more ports of multiple devices. By default, each bridge domain maintains its own forwarding database of MAC addresses learned from packets received on ports belonging to that bridge domain. See also *broadcast domain* and *VLAN*.
- B-TAG—A field defined in the IEEE 802.1ah provider MAC encapsulation header that carries the backbone VLAN identifier information. The format of the B-TAG field is the same as that of the IEEE 802.1ad S-TAG field. See also *S-TAG*.
- B-VID—The specific VLAN identifier carried in a B-TAG.
- CIST—Common and Internal Spanning Tree. The single spanning tree calculated by the spanning tree protocol (STP) and the rapid spanning tree protocol (RSTP) and the logical continuation of that connectivity through multiple spanning tree (MST) bridges and regions, calculated to ensure that all LANs in the bridged LAN are simply and fully connected. See also *MSTI*.
- Ethernet—A term loosely applied to a family of LAN standards based on the original proprietary Ethernet from DEC, Intel, and Xerox (DIX Ethernet), and the open specifications developed by the IEEE 802.3 committee (IEEE 802.3 LANs). In practice, few LANs comply completely with DIX Ethernet or IEEE 802.3.
- IRB—Integrated bridging and routing. IRB provides simultaneous support for Layer 2 (L2) bridging and Layer 3 (L3) routing within the same bridge domain. Packets arriving on an interface of the bridge domain are L2 switched or L3

routed based on the destination MAC address. Packets addressed to the router's MAC address are routed to other L3 interfaces.

- I-SID—The 24–bit service instance identifier field carried inside an I-TAG. The I-SID defines the service instance to which the frame is mapped.
- I-TAG—A field defined in the IEEE 802.1ah provider MAC encapsulation header that carries the service instance information (I-SID) associated with the frame.
- Learning domain—A MAC address database where the MAC addresses are added based on the normalized VLAN tags.
- LFM—Link fault management. A method used to detect problems on links and spans on an Ethernet network defined in IEEE 802.3ah. See also *OAM*.
- MSTI—Multiple Spanning Tree Instance. One of a number of spanning trees calculated by MSTP within an MST region. The MSTI provides a simple and fully connected active topology for frames classified as belonging to a VLAN that is mapped to the MSTI by the MST configuration table used by the MST bridges of that MST region. See also *CIST*.
- MSTP—Multiple Spanning Tree Protocol. A spanning-tree protocol used to prevent loops in bridge configurations. Unlike other types of STPs, MSTP can block ports selectively by VLAN. See also *RSTP*.
- OAM—Operation, Administration, and Maintenance. A set of tools used to provide management for links, device, and networks. See also *LFM*.
- PBB—Provider backbone bridge.
- PBBN—Provider backbone bridged network.
- Q-in-Q—See *802.1ad*.
- RSTP—Rapid Spanning Tree Protocol. A spanning-tree protocol used to prevent loops in bridge configurations. RSTP is not aware of VLANs and blocks ports at the physical level. See also *MSTP*.
- S-TAG—A field defined in the IEEE 802.1ad Q-in-Q encapsulation header that carries the S-VLAN identifier information. See also *B-TAG*.
- S-tagged service interface—The interface between a customer edge (CE) device and the I-BEB or IB-BEB network components. Frames passed through this interface contain an S-TAG field. See also *B-tagged service interface*.
- S-VLAN—The specific service instance VLAN identifier carried inside the S-TAG field. See also *B-VID*.
- Switch—A network device that attempts to perform as much of the forwarding task in hardware as possible. The switch can function as a bridge (LAN switch), router, or some other specialized device, and forwards frames, packets, or other data units. See also *bridge*.
- Virtual switch—A routing instance that can contain one or more bridge domains.
- VLAN—Defines a broadcast domain, a set of logical ports that share the same flooding or broadcast characteristics. VLANs span one or more ports on multiple devices. By default, each VLAN maintains its own Layer 2 forwarding database containing MAC addresses learned from packets received on ports belonging to the VLAN. See also *bridge domain*.

At this point, these acronyms and terms are just a bewildering array of letters and words. It is the goal of this manual to make the contents of this list familiar and allow you to place each of them in context and understand how they relate to each other. To do that, a basic understanding of modern Ethernet standards and technology is necessary.

Networking and Internetworking with Bridges and Routers

Traditionally, different hardware, software, and protocols have been used on LANs and on networks that cover wider areas (national or global). A LAN switch is different than a router, an Ethernet frame is different than an IP packet, and the methods used to find destination MAC addresses are different than those used to find destination IP addresses. This is because LANs based on Ethernet were intended for different network environments than networks based on IP. The Internet protocol suite (TCP/IP) was intended as an internetworking method to connect local customer networks. The local customer network that a service provider's IP routers connected was usually based on some form of Ethernet. This is why Ethernet and IP fit so well together: Ethernet defines the LAN, and the Internet protocols define how these LANs are connected.

More specifically, Ethernet LANs and IP networks occupy different layers of the Internet's TCP/IP protocol suite. Between sender and receiver, networks deal with the bottom three layers of the model: the physical layer (L1), the data link or MAC layer (L2), and the network layer (L3).

NOTE: These layers are also found in the Open Systems Interconnect Reference Model (OSI-RM); however, in this chapter they are applied to the TCP/IP protocol suite.

All digital networks ultimately deal with zeroes and ones, and the physical layer defines bit representation on the media. Physical layer standards also define mechanical aspects of the network, such as electrical characteristics or connector shapes, functional aspects such as bit sequence and organization, and so on. The physical layer only “spits bits” and has very little of the intelligence required to implement a complete network. Devices that connect LAN segments at the physical layer are called *hubs*, and all bits that appear on one port of the hub are also sent out on the other ports. This also means that bad bits that appear on one LAN segment are propagated to all other LAN segments.

Above the physical layer, the data link layer defines the first-order bit structure, or *frame*, for the network type. Also loosely called the MAC layer (technically, the MAC layer is a sublayer required only on LANs), L2 sends and receives frames. Frames are the last things that bits were before they left the sender and the first things that bits become when they arrive on an interface. Because frames have a defined structure, unlike bits, frames can be used for error detection, control plane activities (not all frames must carry user data: some frames are used by the network to control the link), and so forth. LAN segments can be linked at the frame level, and these devices are called *bridges*. Bridges examine arriving frames and decide whether to forward them on an interface. All bridges today are called *learning bridges* because they can find out more about the network than could older bridges that were less intelligent devices. Bridges learn much about the LAN segments they connect to from protocols like those in the Spanning Tree Protocol (STP) family.

The network layer (L3) is the highest layer used by network nodes to forward traffic as part of the data plane. On the Internet, the network layer is the IP layer and can run either IPv4 or IPv6, which are independent implementations of the same functions. The IP layer defines the structure and purpose of the packet, which is in turn the content of the frame at L2. As expected, LAN segments (which now form perfectly functional networks on their own at the frame level) can be linked at the network layer, and in fact that is one of the major functions of IP. Devices that link LANs at the network layer are called *routers*, and IP routers are the network nodes of the Internet.

Addresses at L2 and L3

The Internet is a global, public network with IP subnets connected by routers and exchanging packets. Can a global, public network consist of Ethernet LANs connected by bridges and exchanging frames? Yes, it can, but there are several differences that must be addressed before Ethernet can function as effectively as IP in the metropolitan area (Metro Ethernet), let alone globally. One of the key differences is the addresses used by L2 frames and L3 packets.

Both Ethernet and IP use globally unique network addresses that can be used as the basis for a truly global network. Ethernet MAC addresses come from the IEEE and IP subnet addresses come from various Internet authorities. (IP also employs a naming convention absent in Ethernet, but we'll ignore that in this discussion.) The key differences in how these addresses are assigned make all the difference when it comes to the basic functions of a bridge as opposed to a router.

NOTE: The opposite of a “globally unique network address” is the “locally significant connection identifier” which connects two endpoints on a network. For example, MPLS labels such as 1000001 can repeat in a network, but an IP address such as 192.168.27.48 can appear on the Internet in only one place at a time (otherwise it is an error).

All devices on LANs that are attached to the Internet have both MAC layer and IP addresses. Frames and packets contain both source and destination addresses in their headers. In general:

- MAC addresses are 48 bits long. The first 24 bits are assigned by the IEEE and form the organizationally unique identifier (OUI) of the manufacturer or vendor requesting the address. The last 24 bits form the serial number of the LAN interface cards and their uniqueness must be enforced by the company (some companies reuse numbers of bad or returned cards while others do not).
- IPv4 addresses are 32 bits long. A variable number of the beginning bits are assigned by an Internet authority and represent a subnet located somewhere in the world. The remaining bits are assigned locally and, when joined to the network portion of the address, uniquely identify some host on a particular network.

- IPv6 addresses are 128 bits long. Although there are significant differences, for the purposes of this discussion, it is enough to point out that there is also a network and host portion to an IPv6 address.

Note that MAC addresses are mainly organized by manufacturer and IP addresses are organized by network, which is located in a particular place. Therefore, the IP address can easily be used by routers for a packet's overall direction (for example, "192.168.27.48 is west of here"). However, the MAC addresses on a vendor's interface cards can end up anywhere in the world, and often do. Consider a Juniper Networks router as a simple example. Every Ethernet LAN interface on the router that sends or receives packets places them inside Ethernet frames with MAC addresses. All of these interfaces share the initial 24 bits assigned to Juniper Networks. Two might differ only in one digit from one interface to another. Yet the routers containing these MAC interfaces could be located on opposite sides of the world.

An Internet backbone router only needs a table entry for every network (not host) in the world. Most other routers only have a portion of this full table, and a default route for forwarding packets with no entries in their table. In contrast, to perform the same role, a bridge would need one table entry for every LAN interface, on host or bridge, in the world. This is hard enough to do for Ethernets that span a metropolitan area, let alone the entire world.

NOTE: There are other reasons that Ethernet would be hard-pressed to become a truly global network, including the fact that MAC addresses do not often have names associated with them while IP addresses do (for example, 192.168.27.48 might be host48.accounting.juniper.net). This section addresses only the address issues.

The Benefits of Ethernet

In spite of the difficulties of using a bridge to perform the network role of a router, many vendors, customers, and service providers are attracted to the idea of using Ethernet in as many places of their networks as possible. The perceived benefits of Ethernet are:

- Most information starts and ends inside Ethernet frames. Today, this applies to data, as well as voice (for example, VoIP) and video (for example, Web cams).
- Ethernet frames have all the essentials for networking, such as globally unique source and destination addresses, error control, and so on.
- Ethernet frames can carry any kind of packet. Networking at Layer 2 is protocol independent (independent of the Layer 3 protocol). Layer 2 networks work for IP packets and all other Layer 3 protocols.
- More layers added to the Ethernet frame only slow the networking process down ("nodal processing delay").
- Adjunct networking features such as class of service (CoS) or multicasting can be added to Ethernet as readily as IP networks.

If more of the end-to-end transfer of information from a source to a destination can be done in the form of Ethernet frames, more of the benefits of Ethernet can be realized on the network. Networking at Layer 2 can be a powerful adjunct to IP networking, but it is not usually a substitute for IP networking.

NOTE: Networking at the frame level says nothing about the presence or absence of IP addresses at the packet level. Almost all ports, links, and devices on a network of LAN switches still have IP addresses, just as do all the source and destination hosts. There are many reasons for the continued need for IP, not the least of which is the need to manage the network. A device or link without an IP address is usually invisible to most management applications. Also, utilities such as remote access for diagnostics, file transfer of configurations and software, and so on cannot run without IP addresses as well as MAC addresses.

Handling MAC Addresses

If a networked L2 device such as a bridge or LAN switch could contain a list of all known MAC addresses, then the network node could function in much the same way as a router, forwarding frames instead of packets hop-by-hop through the network from source LAN to destination LAN. However, the MAC address is much larger than the IPv4 address currently used on the Internet backbone (48 bits compared to the 32 bits of IPv4). This poses problems. Also, because the MAC address has no "network organization" like the IPv4 or IPv6 address, an L2 network node must potentially store every conceivable MAC address in memory for next-hop table lookups. Instead of tables of about 125,000 entries, every L2 network node would have to store millions of entries (for example, 24 bits, the potential NIC production from *one* Ethernet vendor, would require a table of more than 16 million entries).

MAC Addresses, VLAN Tags, and Forwarding

VLAN tags were not developed as a way to limit network node table entries. They were originally invented to allow LAN switches to distinguish between physical groups of LAN ports and logical groups of LAN ports. In other words, there was a need to configure a LAN switch (or group of local LAN switches) to know that "these ports belong to VLAN A" and "these ports belong to VLAN B."

This was important because of how all LANs, not just Ethernet, work at the frame level. Lots of frames on a LAN are broadcast to all stations (hosts and network nodes) on the LAN segment. Also, multicasting works by flooding traffic within the VLAN. The stations that received broadcast frames form the *broadcast domain* of the LAN. Only Ethernet frames belonging to same broadcast domain are forwarded out certain ports on the LAN switch. This prevents broadcast storms and isolates routine control frames onto the LAN segment where they make the most sense.

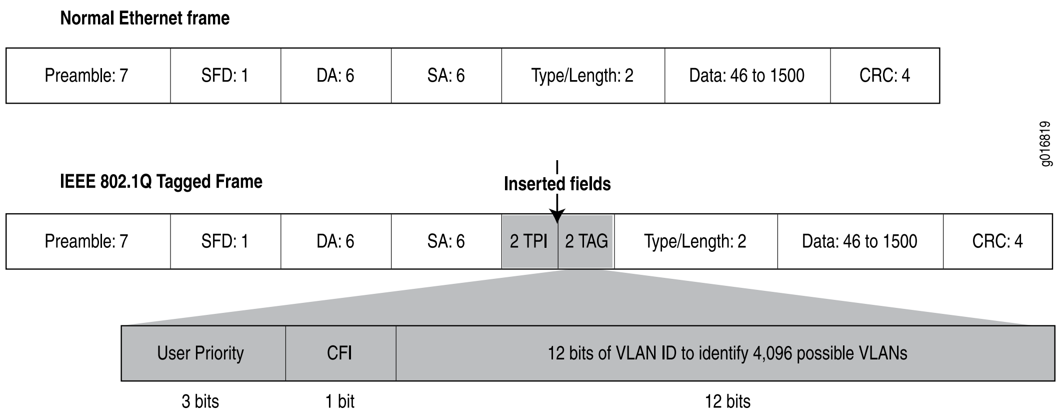
The VLAN tag was invented to distinguish among different VLAN broadcast domains on a group of LAN switches. The VLAN tag is a two-byte field inserted between the source MAC address and the Ethertype (or length) field in an Ethernet frame. Another two-byte field, the Tag Protocol Identifier (TPI or TPID), precedes the VLAN tag field.

Two fields were necessary to hold one piece of information, the VLAN tag, to enable receivers to distinguish between untagged or plain Ethernet frames and those containing VLAN tags. A mechanism was required to differentiate between the

Ethertype and length field for the untagged case and to distinguish among VLAN tag, Ethertype, and length field for the tagged case. The answer was to constrain the TPID field to values that were not valid Ethernet frame lengths or defined as valid Ethertypes. The first VLAN tag added to an Ethernet frame is always indicated by a TPID value of 0x8100. This is not the VLAN identifier, which appears in the next two bytes.

In Figure 1 on page 10, a native or normal Ethernet frame is compared to a VLAN-tagged Ethernet frame. The lengths of each field, in bytes, is shown next to the field name.

Figure 1: Native (Normal) and VLAN-Tagged Ethernet Fames



The VLAN tag subtracts four bytes from the total MTU length of the Ethernet frame, but this is seldom a problem if kept in mind. When this tag is used in an Ethernet frame, the frame complies with the IEEE 802.1Q (formerly IEEE 802.1q) specification.

Together, the four added bytes form the VLAN tag, but the individual fields that comprise it are more important. The 2–byte TPID field is just a number and has no structure, only having allowed and disallowed values. However, the 2-byte Tag Control Information (TCI) field has a defined structure:

- The three bits of the User Priority field are defined by the IEEE 802.1p specification. These can mimic class-of-service (CoS) parameters established at other layers of the network (IP precedence bits, or MPLS EXP bits, and so on).
- The Canonical Format Indicator (CFI) bit indicates whether the following 12 bits of VLAN identifier conform to Ethernet or not. For Ethernet frames, this bit is always set to 0. (The other possible value, CFI = 1, is used for Token Ring LANs, and tagged frames should never be bridged between an Ethernet and Token Ring LAN regardless of the VLAN tag or MAC address.)
- The 12-bit VLAN ID allows for 4096 possible VLANs, but not all values are used in all cases.

Nesting VLAN Tags

The use of VLAN tagging to group (or bundle) sets of MAC addresses is a start toward a method of forwarding LAN traffic based on information found in the frame, not on IP address in the packet. However, there is a major limitation in trying to build forwarding tables based on VLAN tags. Simply put, there are not enough VLAN tags.

Twelve bits only supply enough space for 4096 unique VLAN tags. This is hardly enough for all the LANs on a large corporate campus, let alone the whole world. A 12-bit tag might suffice for the local campus arena, but for the metropolitan area, comprising a whole city, more bits are needed.

The number of bits in the VLAN tag, two bytes for the TPID and two bytes for the TCI field, are fixed and cannot be extended. However, another VLAN tag can be added to the frame, forming an inner and outer VLAN tag arrangement. This arrangement is defined in the IEEE 802.1ad specification and applies to devices that function on the provider bridge level. This means that Ethernet frames tagged at the local (or customer) VLAN level can receive another outer VLAN tag when they are sent to the provider's LAN switches. As a result, Ethernet frames can be switched across a metropolitan area, not just among the local organizations devices at the campus level.

The outer tag defined in IEEE 802.1ad is often called the Virtual Metropolitan Area Network (VMAN) tag, a good way to recall the intended scope of the specification. The outer tag is placed after the MAC source address, moving the inner tag backwards in the frame. Both tags can be added at the same time by the same device (called a push/push operation), changed by a device (a swap operation), or removed by a device one at a time (pop) or together (pop/pop). Devices can perform elaborate variations on these operations (such as pop/swap/push) to accomplish the necessary networking tasks with the frames they process.

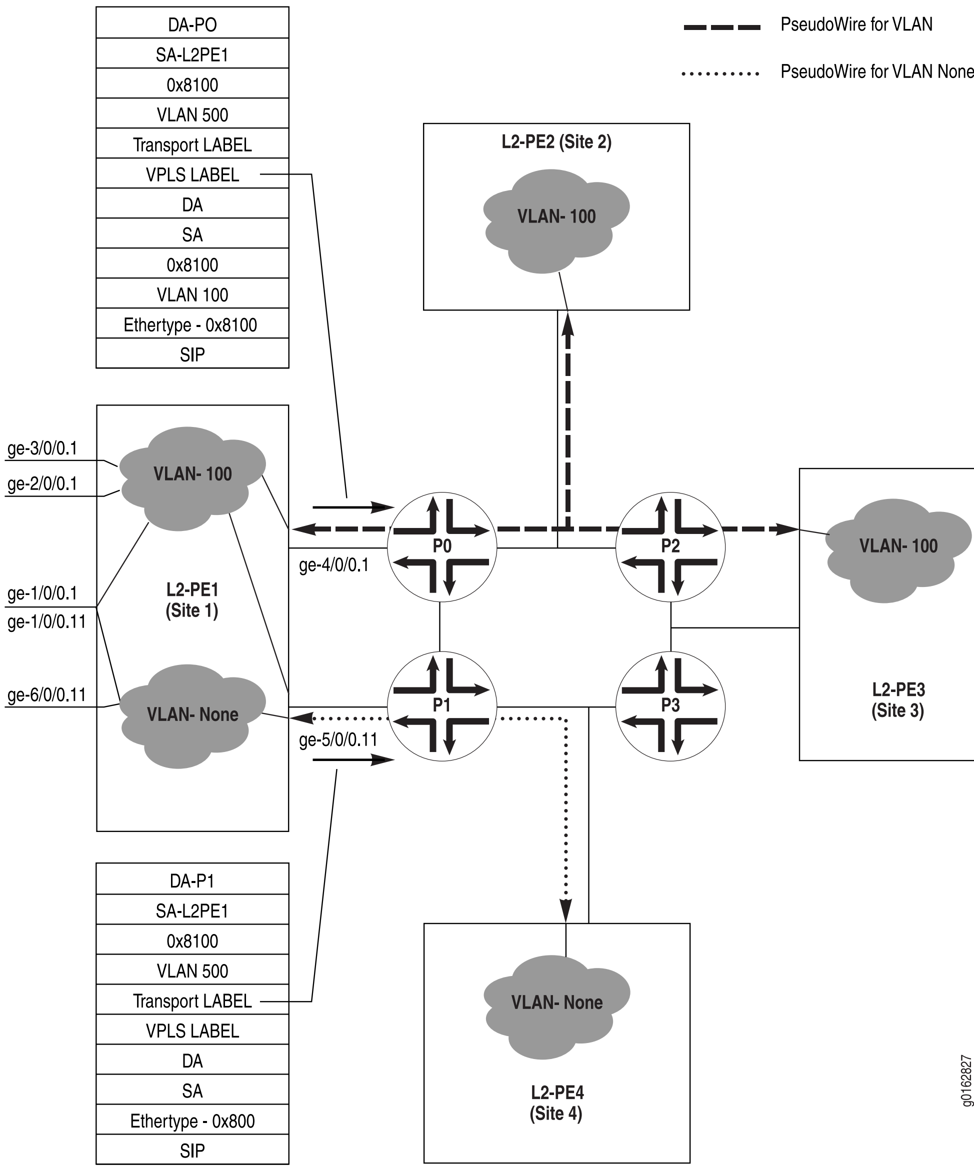
The IEEE specification indicates that the outer tag of a doubly-tagged Ethernet frame should have a TPID value of 0x88a8. Any network device can easily tell if it has received a frame with one tag (0x8100) or two tags (0x88a8). However, because the value 0x8100 always means that a VLAN tag is present, most vendors and networks use the same TPID value (0x8100) for the inner and outer tags. As long as the configuration and processing are consistent, there is no confusion, and the TPID value can usually be changed if necessary.

How do nested VLAN tags solve the VLAN numbering limitation? Taken together, the two VLAN tags can be thought of as providing 24 bits for tagging space: 12 bits at the outer level and 12 bits at the inner level. However, it is important to realize that the bits are not acted on as if they were all one tag. Even when the tags are nested, bridges on a provider backbone will normally only switch on the outer VLAN tag. All in all, the inner 12-bit tagging space is more than adequate for a Metro Ethernet network. Any limitations in the VLAN tag space can be addressed by adding more VLAN tags to the basic Ethernet frame.

Metro Ethernet Network

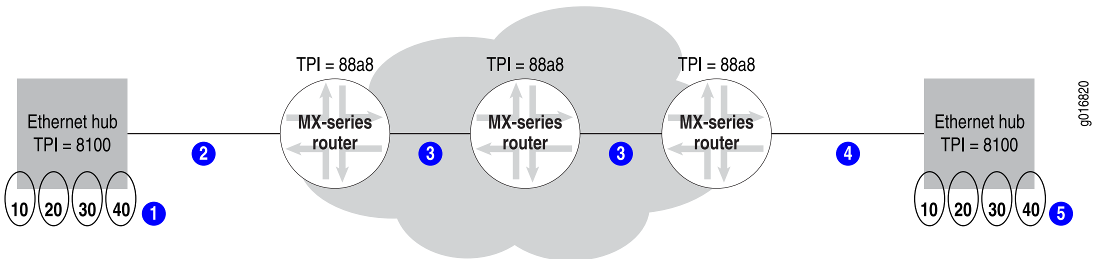
What would a Metro Ethernet network with MX-series routers look like? It is very likely that the Metro Ethernet network will place MX-series routers at the edge of a VPLS and MPLS core network. The VLAN labels in the packet are stacked with MPLS labels, as shown in Figure 2 on page 12. For a more detailed examination of this type of Metro Ethernet network, see “VPLS Labels and VLAN Tags” on page 37.

Figure 2: A Metro Ethernet Network



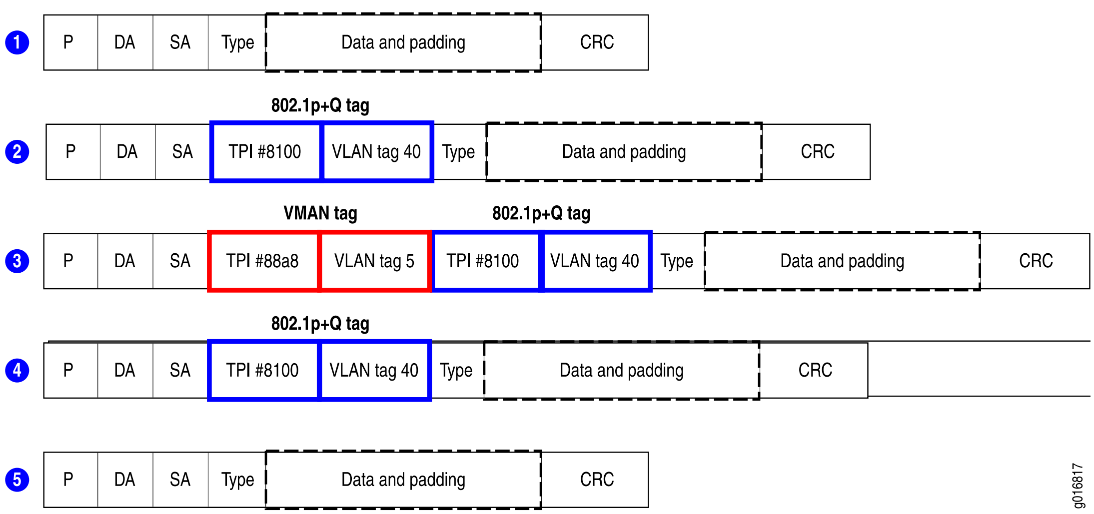
Another possible configuration, this one without the VPLS and MPLS core, is shown in Figure 3 on page 13.

Figure 3: A Metro Ethernet Network with MX-series Routers



In Figure 3 on page 13, the circled numbers reflect the different formats that the Ethernet frames can take as the frames make their way from a host on one Ethernet switching hub to a host on the other hub. The frame can have two VLAN tags (inner and outer), one tag (only the inner), or no tags at all. The structure of these various Ethernet frames is shown in Figure 4 on page 13.

Figure 4: VLAN Tags on a Metro Ethernet Network



As the frame flows from a LAN-based host on one end of Figure 4 on page 13 to the other, the Ethernet frame can have:

- No VLAN tags—At locations 1 and 5, the Ethernet frames can be native and have no VLAN tags at all (many NIC cards can include configuration of a VLAN identifier, but not all).
- One VLAN tag—At locations 2 and 4, from the VLAN-aware switching hub to the MX-series router, the Ethernet frame has one VLAN tag (if a VLAN tag is not present on arriving frames, a tag is added by the MX-series router).
- Two VLAN tags—At location 3, between two provider bridges, the MX-series routers exchange frames with two VLAN tags. The outer tags are added and removed by the MX-series routers.

Part 2

Solutions for MX-series

Chapter 2

Configuring Basic MX-series Layer 2 Features

You configure MX-series routers exactly as you would any other router running the JUNOS software. That is, all the familiar Layer 3 (L3) features and protocols are available on the MX-series routers. However, you can configure Layer 2 (L2) features that are unique to the MX-series routers. This chapter addresses L2 configuration for the MX-series routers. For information about configuring L3 features and protocols, as well as comprehensive information about interfaces and system basics, please see the other JUNOS configuration guides.

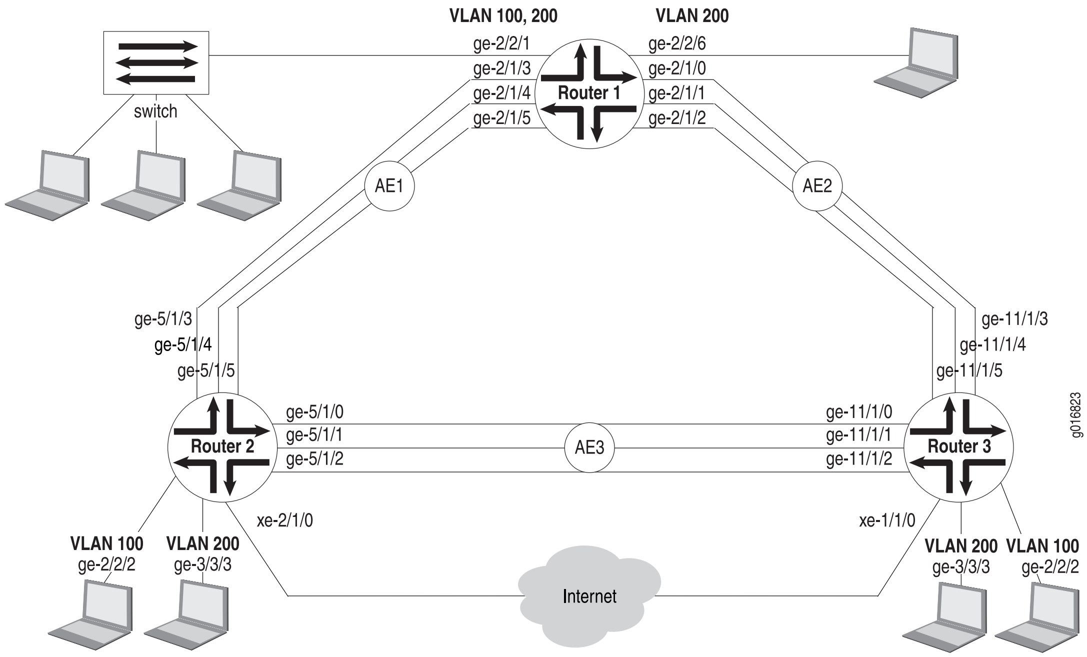
Configuring L2 features on an MX-series router can vary from the very simple (aggregated Ethernet trunk interfaces, spanning trees), to the more complex (inner and outer VLAN tags, broadcast domains), to the very complicated (integrated bridging and routing, L2 filtering). This chapter offers a fairly complex configuration for L2 processing in a bridged environment.

Generally, there are four things that you must configure in an L2 environment:

- Interfaces and virtual LAN (VLAN) tags—L2 interfaces are usually various type of Ethernet links with VLAN tags used to connect to customer devices or other bridges or routers.
- Bridge domains and virtual switches—Bridge domains limit the scope of media access control (MAC) learning (and thereby the size of the MAC table) and also determine where the device should propagate frames sent to broadcast, unknown unicast, and multicast (BUM) MAC addresses. Virtual switches allow for the configuration of multiple, independent bridge domains.
- Spanning Tree Protocols (xSTP, where the “x” represents the STP type)—Bridges function by associating a MAC address with an interface, similar to the way a router associates an IP network address with a next-hop interface. Just as routing protocols use packets to detect and prevent routing loops, bridges use xSTP frames to detect and prevent bridging loops. (L2 loops are more devastating to a network because of the broadcast nature of Ethernet LANs.)
- Integrated bridging and routing (IRB)—Support for both Layer 2 bridging and Layer 3 routing on the same interface. Frames are bridged if they are not sent to the router's MAC address. Frames sent to the router's MAC address are routed to other interfaces configured for Layer 3 routing.

Consider the network in Figure 5 on page 18. The figure shows three MX-series routers acting as L2 devices.

Figure 5: Bridging Network with MX-series Routers



The network administrator wants to configure these links and devices so that:

- The six Gigabit Ethernet links between Router 1 and the other routers (ge-2/1/0 through ge-2/1/5) are gathered into two Aggregated Ethernet (AE) links mixing bridged traffic from the VLANs. AE1 will consist of the first three links and AE2 will use the last three links. The same approach is taken for the links on Router 2 and Router 3.
- The Gigabit Ethernet links from Router 1 to the customer devices (ge-2/2/1 and ge-2/2/6) will be bridged and include VLAN tag 100 on ge-2/2/1 and VLAN tag 200 on ge-2/2/6. The other two routers, Router 2 and Router 3, also have two ports configured to handle VLAN 100 on one port (ge-2/2/2) and VLAN 200 on the other (ge-3/3/3).
- The routers have bridge domains reflecting these VLAN configurations.
- Because the VLANs appear on each MX-series router, the routers run Multiple STP (MSTP) on the links connecting them to prevent bridging loops (Rapid STP, or RSTP, does not recognize VLAN tags and blocks ports without regard for VLAN tagging).
- Router 2 and Router 3 have IRB configured so that they can pass traffic to other routers in the rest of the network. These interfaces are configured in “Configuring Integrated Bridging and Routing” on page 26.

This chapter provides the following information about this MX-series L2 configuration of the three routers:

Configuring the Interfaces and VLAN Tags

Configure the Ethernet interfaces and VLAN tags on all three routers.

NOTE: The configurations in this chapter are only partial examples of complete and functional router configurations. Do not copy these configurations and use them directly on an actual system.

Router 1 On Router 1, configure the Ethernet interfaces and VLAN tags:

```
[edit chassis]
aggregated-devices {
   ethernet {
      device-count 2; # Number of AE interfaces on router
   }
}
[edit]
interfaces ge-2/1/0 {
   gigether-options {
      802.3ad ae2;
   }
}
interfaces ge-2/1/1 {
   gigether-options {
      802.3ad ae2;
   }
}
interfaces ge-2/1/2 {
   gigether-options {
      802.3ad ae2;
   }
}
interfaces ge-2/1/3 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ge-2/1/4 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ge-2/1/5 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ge-2/2/1 {
   encapsulation flexible-ethernet-services;
   vlan-tagging; # Customer interface uses singly-tagged frames
```

```
    unit 100 {
        encapsulation vlan-bridge;
        vlan-id 100;
    }
    unit 200 {
        encapsulation vlan-bridge;
        vlan-id 200;
    }
}
interfaces ge-2/2/6 {
    encapsulation flexible-ethernet-services;
    vlan-tagging; # Customer interface uses singly-tagged frames
    unit 200 {
        encapsulation vlan-bridge;
        vlan-id 200;
    }
}
interfaces ae1 {
    encapsulation extended-vlan-bridge;
    vlan-tagging;
    unit 100 {
        vlan-id 100;
    }
    unit 200 {
        vlan-id 200;
    }
}
interfaces ae2 {
    encapsulation extended-vlan-bridge;
    vlan-tagging;
    unit 100 {
        vlan-id 100;
    }
    unit 200 {
        vlan-id 200;
    }
}
```

Router 2 On Router 2, configure the Ethernet interfaces and VLAN tags:

```
[edit chassis]
aggregated-devices {
    ethernet {
        device-count 2; # Number of AE interfaces on the router
    }
}

[edit]
interfaces ge-2/2/2 {
    encapsulation flexible-ethernet-services;
    vlan-tagging; # Customer interface uses singly-tagged frames
    unit 100 {
        encapsulation vlan-bridge;
        vlan-id 100;
    }
}
```

```
interfaces ge-3/3/3 {
   encapsulation flexible-ethernet-services;
   vlan-tagging; # Customer interface uses singly-tagged frames
   unit 200 {
      encapsulation vlan-bridge;
      vlan-id 200;
   }
}
interfaces ge-5/1/0 {
   gigether-options {
      802.3ad ae3;
   }
}
interfaces ge-5/1/1 {
   gigether-options {
      802.3ad ae3;
   }
}
interfaces ge-5/1/2 {
   gigether-options {
      802.3ad ae3;
   }
}
interfaces ge-5/1/3 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ge-5/1/4 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ge-5/1/5 {
   gigether-options {
      802.3ad ae1;
   }
}
interfaces ae1 {
   encapsulation extended-vlan-bridge;
   vlan-tagging;
   unit 100 {
      vlan-id 100;
   }
   unit 200 {
      vlan-id 200;
   }
}
interfaces ae3 {
   encapsulation extended-vlan-bridge;
   vlan-tagging;
   unit 100 {
      vlan-id 100;
   }
   unit 200 {
      vlan-id 200;
```

```
    }
}
```

Router 3 On Router 3, configure the Ethernet interfaces and VLAN tags:

```
[edit chassis]
aggregated-devices {
    ethernet {
        device-count 2; # Number of AE interfaces on router
    }
}

[edit]
interfaces ge-2/2/2 {
    encapsulation flexible-etherent-services;
    vlan-tagging; # Customer interface uses singly-tagged frames
    unit 100 {
        encapsulation vlan-bridge;
        vlan-id 100;
    }
}
interfaces ge-3/3/3 {
    encapsulation flexible-ethernet-services;
    vlan-tagging; # Customer interface uses singly-tagged frames
    unit 200 {
        encapsulation vlan-bridge;
        vlan-id 200;
    }
}
[edit]
interfaces ge-11/1/0 {
    gigether-options {
        802.3ad ae3;
    }
}
interfaces ge-11/1/1 {
    gigether-options {
        802.3ad ae3;
    }
}
interfaces ge-11/1/2 {
    gigether-options {
        802.3ad ae3;
    }
}
interfaces ge-11/1/3 {
    gigether-options {
        802.3ad ae2;
    }
}
interfaces ge-11/1/4 {
    gigether-options {
        802.3ad ae2;
    }
}
interfaces ge-11/1/5 {
```

```
    gigether-options {
        802.3ad ae2;
    }
}
interfaces ae2 {
    encapsulation extended-vlan-bridge;
    vlan-tagging;
    unit 100 {
        vlan-id 100;
    }
    unit 200 {
        vlan-id 200;
    }
}
interfaces ae3 {
    encapsulation extended-vlan-bridge;
    vlan-tagging;
    unit 100 {
        vlan-id 100;
    }
    unit 200 {
        vlan-id 200;
    }
}
```

Configuring Virtual Switches and Bridge Domains

Configure the virtual switches and bridge domains on all three routers. There is always a default virtual switch in the router for L2 functions; however, if there is only one L2 network, then the virtual switch instance type is not needed.

Router 1 Configure a bridge domain on Router 1:

```
[edit]
bridge-domains {
    vlan100 {
        domain-type bridge;
        vlan-id 100;
        interface ge-2/2/1.100;
        interface ae1.100;
        interface ae2.100;
    }
    vlan200 {
        domain-type bridge;
        vlan-id 200;
        interface ge-2/2/1.200;
        interface ge-2/2/6.200;
        interface ae1.200;
        interface ae2.200;
    }
}
```

Router 2 Configure a bridge domain on Router 2:

```
[edit]
bridge-domains {
    vlan100 {
```

```
        domain-type bridge;
        vlan-id 100;
        interface ge-2/2/2.100;
        interface ae1.100;
        interface ae3.100;
    }
    vlan200 {
        domain-type bridge;
        vlan-id 200;
        interface ge-3/3/3.200;
        interface ae1.200;
        interface ae3.200;
    }
}
```

Router 3 Configure a broadcast domain on Router 3:

```
[edit]
bridge-domains {
    vlan100 {
        domain-type bridge;
        vlan-id 100;
        interface ge-2/2/2.100;
        interface ae2.100;
        interface ae3.100;
    }
    vlan200 {
        domain-type bridge;
        vlan-id 200;
        interface ge-3/3/3.200;
        interface ae2.200;
        interface ae3.200;
    }
}
```

Configuring Spanning Tree Protocols

Configure the Spanning Tree Protocol on all three routers. This is necessary to avoid the potential bridging loop formed by the triangular architecture of the routers. MSTP is configured on the three routers so the set of VLANs has an independent, loop-free topology. The Layer 2 traffic can be load-shared over 65 independent paths (64 Multiple Spanning Tree Instances [MSTIs] and one Common and Internal Spanning Tree [CIST]), each spanning a set of VLANs. The configuration names, revision level, and VLAN-to-MSTI mapping must match in order to utilize the load-sharing capabilities of MSTP (otherwise, each router will be in a different region).

Router 1 Configure MSTP on Router 1:

```
[edit]
protocols {
    mstp {
        configuration-name mstp-for-R1-2-3; # The names must match to be in the same
            region
        revision-level 3; # The revision levels must match
        bridge-priority 0; # This bridge acts as root bridge for VLAN 100 and 200
        interface ae1;
```

```
        interface ae2;
        msti 1 {
            vlan100; # This VLAN corresponds to MSTP instance 1
        }
        msti 2 {
            vlan200; # This VLAN corresponds to MSTP instance 2
        }
    }
}
```

Router 2 Configure MSTP on Router 2:

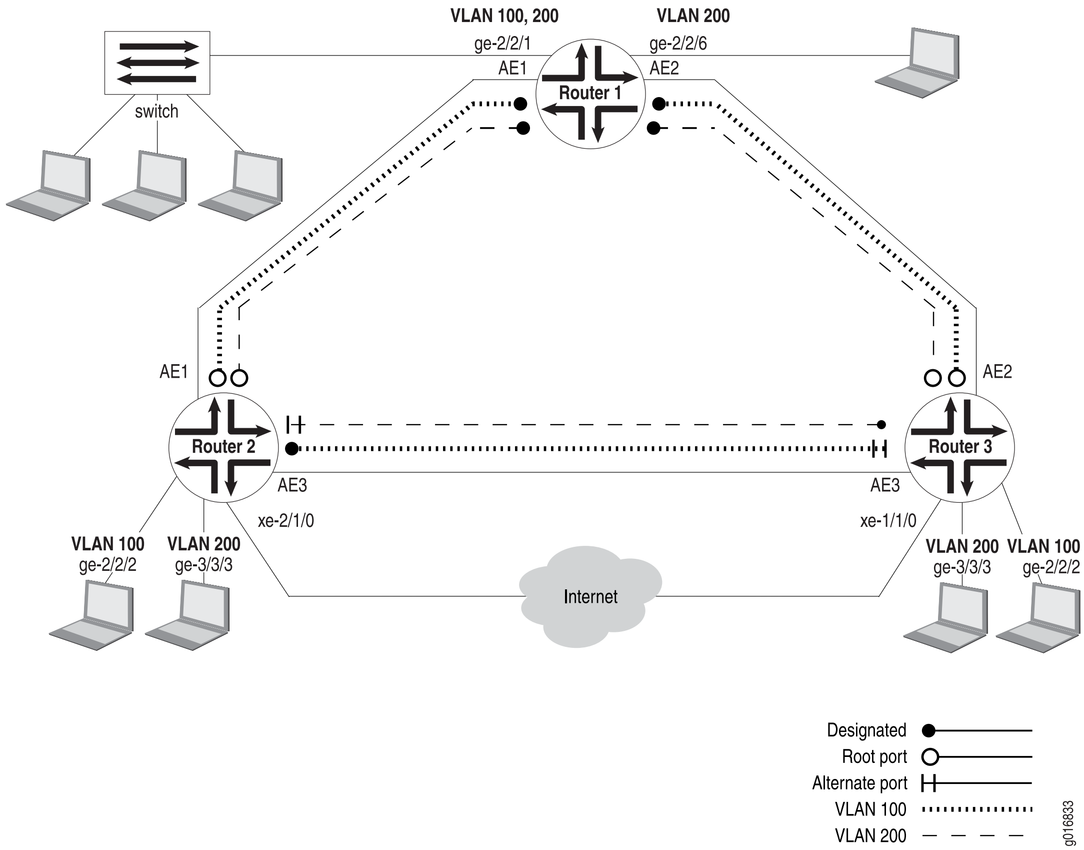
```
[edit]
protocols {
    mstp {
        configuration-name mstp-for-R1-2-3; # The names must match to be in the same
            region
        revision-level 3; # The revision levels must match
        interface ae1;
        interface ae3;
        msti 1 {
            vlan100; # This VLAN corresponds to MSTP instance 1
            bridge-priority 4096; # This bridge acts as VLAN 100 designated bridge on
                                  # the R2-R3 segment
        }
        msti 2 {
            vlan200; # This VLAN corresponds to MSTP instance 2
        }
    }
}
```

Router 3 Configure MSTP on Router 3:

```
[edit]
protocols {
    mstp {
        configuration-name mstp-for-R1-2-3; # The names must match to be in the same
            region
        revision-level 3; # The revision levels must match
        interface ae2;
        interface ae3;
        msti 1 {
            vlan100; # This VLAN corresponds to MSTP instance 1
        }
        msti 2 {
            vlan200; # This VLAN corresponds to MSTP instance 2
            bridge-priority 4096; # This bridge acts as VLAN 200 designated bridge on
                                  # the R2-R3 segment
        }
    }
}
```

As a result of this configuration, VLAN 100 and VLAN 200 share physical links, but have different designated ports, root ports, and alternate ports on the three different routers. The designated, root, and alternate ports for the two VLANs on the three routers are shown in Figure 6 on page 26.

Figure 6: Designated, Root, and Alternate Ports



Configuring Integrated Bridging and Routing

Router 2 and Router 3 on the bridging network act as a kind of gateway to the L3 routers in the rest of the network. Router 2 and Router 3 must be able to route packets as well as bridge frames. This requires the configuration of integrated routing and bridging (IRB) on Routers 2 and 3. The link to the router network is xe-2/1/0 on Router 2 and xe-1/1/0 on Router 3.

You configure IRB in two steps:

1. Configure the IRB interface using the irb statement.
2. Reference the IRB interface at the bridge domain level of the configuration.

IRB supports Layer 2 bridging and Layer 3 routing on the same interface. If the MAC address on the arriving frame is the same as that of the IRB interface, then the packet inside the frame is routed. Otherwise, the MAC address is learned or looked up in the MAC address database.

Configure IRB on Router 2 and Router 3. The Virtual Router Redundancy Protocol (VRRP) is configured on the IRB interface so that both links can be used to carry traffic between the bridge domain and the router network.

Router 2 Configure the router link and IRB:

```
[edit interfaces]
xe-2/1/0 {
   unit 0 {
      family inet {
         address 10.0.10.2/24; # Routing interface
      }
   }
}
irb {
   unit 0 {
      family inet {
         address 10.0.1.2/24 {
         vrrp-group 1 {
            virtual-address 10.0.1.51;
               priority 254;
            }
         }
      }
   }
   unit 1 {
      family inet {
         address 10.0.2.2/24 {
         vrrp-group 2 {
            virtual-address 10.0.2.51;
               priority 100;
            }
         }
      }
   }
}

[edit]
bridge-domains {
   vlan-100 {
      domain-type bridge;
      vlan-id 100;
      interface ge-2/2/2.100;
      interface ae1.100;
      interface ae3.100
      routing-interface irb.0;
   }
   vlan-200 {
      domain-type bridge;
      vlan-id 200;
      interface ge-3/3/3.200;
      interface ae1.200;
      interface ae3.200
      routing-interface irb.1;
   }
}
```

Router 3 Configure the router link and IRB:

```
[edit interface]
xe-1/1/0 {
   unit 0 {
      family inet {
         address 10.0.20.3/24; # Routing interface
      }
   }
}
irb {
   unit 0 {
      family inet {
         address 10.0.1.3/24 {
         vrrp-group 1 {
            virtual-address 10.0.1.51;
               priority 100;
            }
         }
      }
   }
   unit 1 {
      family inet {
         address 10.0.2.3/24 {
         vrrp-group 2 {
            virtual-address 10.0.2.51;
               priority 254;
            }
         }
      }
   }
}

[edit]
bridge-domains {
   vlan-100 {
      domain-type bridge;
      vlan-id 100;
      interface ge-2/2/2.100;
      interface ae2.100;
      interface ae3.100;
      routing-interface irb.0;
   }
   vlan-200 {
      domain-type bridge;
      vlan-id 200;
      interface ge-3/3/3.200;
      interface ae2.200;
      interface ae3.200;
      routing-interface irb.1;
   }
}
```

Chapter 3

VLAN Configuration for VPLS and Bridge Domains

This chapter provides configuration and operational information to help you manipulate virtual local area networks (VLANs) within a bridge domain or a virtual private LAN service (VPLS) instance. The VPLS configuration is not covered in this chapter. For more information about configuring Ethernet pseudowires as part of VPLS, see the *JUNOS Software Feature Guide*.

NOTE: This chapter is not intended as a troubleshooting guide. However, you can use it with a broader troubleshooting strategy to identify MX-series router network problems.

The manipulation of VLANs within a bridge domain or a VPLS instance can be done in several ways:

- By using the vlan-map statements at the [edit interfaces] hierarchy level. This chapter does not use vlan-map. For more information about VLAN maps, see the *JUNOS Interfaces Configuration Guide*.
- By using vlan-id statements within a bridge domain or VPLS instance hierarchy. This method is used in the configuration in this chapter.

The vlan-id and vlan-tags statements under the bridge domain or VPLS routing instance are used to:

- Translate (normalize) received VLAN tags, or
- Implicitly create multiple learning domains, each with a “learn” VLAN.

The use of a VLAN map or a normalized VLAN is optional.

NOTE: You cannot use vlan-map when configuring a normalized VLAN.

This chapter discusses the following topics:

VLAN Translation (Normalization)

A packet received on a physical port is only accepted for processing if the VLAN tags of the received packet match the VLAN tags associated with one of the logical interfaces configured on the physical port. The VLAN tags of the received packet are translated only if they are different than the normalized VLAN tags. For the translation case, the vlan-id or vlan-tags statements specify the normalized VLAN. For this case, the terms "learn VLAN" and "normalized VLAN" can be used interchangeably.

Specify the normalized VLAN using one of the following configuration statements:

- vlan-id *vlan-number*
- vlan-id none
- vlan-tags outer *outer-vlan-number* inner *inner-vlan-number*

Configured the normalized VLAN for one of the following scenarios:

Implicit VLAN Translation to a Normalized VLAN

The VLAN tags of a received packet are compared with the normalized VLAN tags specified with either the vlan-id or vlan-tags statements. If the VLAN tags of the received packet are different from the normalized VLAN tags, then appropriate VLAN tag operations (such as push-push, pop-pop, pop-swap, swap-swap, swap, and others) are implicitly made to convert the received VLAN tags to the normalized VLAN tags. For more information about these operations, see the *JUNOS Routing Protocols Configuration Guide*.

Then, the source MAC address of a received packet is learned based on the normalized VLAN configuration.

For output packets, if the VLAN tags associated with an egress logical interface do not match the normalized VLAN tags within the packet, then appropriate VLAN tag operations (such as push-push, pop-pop, pop-swap, swap-swap, swap, and others) are implicitly made to convert the normalized VLAN tags to the VLAN tags for the egress logical interface. For more information about these operations, see the *JUNOS Routing Protocols Configuration Guide*.

Sending Tagged or Untagged Packets over VPLS Virtual Interfaces

If the packets sent over the VPLS virtual interfaces (vt- or lsi- interfaces) need to be tagged by the normalized VLAN, use one of the following configuration statements:

- vlan-id *vlan-number*—Tags all packets sent over the VPLS virtual interface with the configured *vlan-number*. See "VPLS Labels and VLAN Tags" on page 37 for an example of this configuration.
- vlan-tags outer *outer-vlan-number* inner *inner-vlan-number*—Tags all packets sent over the VPLS virtual interfaces with the specified inner and outer VLAN tags.

If the incoming VLAN tags identifying a Layer 2 logical interface are removed when packets are sent over VPLS virtual interfaces, use the vlan-id none statement.

NOTE: Even when the vlan-id none statement is configured, the packets can still contain other customer VLAN tags.

Creating Implicit Learning Domains

Multiple learning domains for a bridge domain or VPLS instance are implicitly created with the vlan-id all statement. This statement provides a mechanism to configure bridging for several VLANs with a minimal amount of configuration and switch resources.

The vlan-id all statement implicitly creates a learning domain for:

- Each inner VLAN (normalized VLAN) of a logical interface with two VLAN tags.
- Each normalized VLAN of a logical interface with one VLAN tag.

A learning domain is a MAC address database where the MAC addresses are added based on the normalized VLAN tags. The normalized VLAN tags associated with a learning domain are always carried within packets sent over VPLS virtual interfaces.

Bridging Packet Flow

Packets received over a Layer 2 logical interface for bridging when a normalized VLAN is configured with either the vlan-id or vlan-tags statements under the bridge domain or the VPLS routing instance are processed with the following steps:

1. A packet received on a physical port is only accepted for further processing if the VLAN tags of the received packet match the VLAN tags associated with one of the logical interfaces configured on that physical port.
2. The VLAN tags of the received packet are compared with the normalized VLAN tags. If the VLAN tags of the received packet are different from the normalized VLAN, then the appropriate VLAN operations (such as push-push, pop-pop, pop-swap, swap-swap, swap, and others) are done implicitly to convert the received VLAN tags to the normalized VLAN tag value. For more information these operations, see the *JUNOS Routing Protocols Configuration Guide*.

3. If the source MAC address of the received packet is not present in the source MAC table, then it is learned based on the normalized VLAN tag value.
4. The packet is forwarded toward one or more egress Layer 2 logical interfaces based on the destination MAC address. A packet with a known unicast destination MAC address is only forwarded to one egress logical interface. For each egress Layer 2 logical interface, the normalized VLAN tag within the packet is compared with the VLAN tags configured on that logical interface. If the VLAN tags associated with an egress logical interface do not match the normalized VLAN tag in the frame, then appropriate VLAN operations (such as push-push, pop-pop, pop-swap, swap-swap, swap, and others) are implicitly done to convert the normalized VLAN tags to the VLAN tags of the egress logical interface. For more information these operations, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring a Normalized VLAN

The following factors are important when configuring a normalized VLAN:

- Use either the vlan-id *vlan-number* statement (to tag all packets with one normalized VLAN tag) or the vlan-tags outer *outer-vlan-number* inner *inner-vlan-number* statement (to tag all packets with the normalized outer and inner VLAN tags) if you want to tag packets sent onto the VPLS pseudowires.
- Use the vlan-id none statement to remove the incoming VLAN tags identifying a Layer 2 logical interface when packets are sent over VPLS pseudowires. This statement is also used to configure shared VLAN learning.

NOTE: The outgoing packets can still contain customer VLAN tags.

- If integrated routing and bridgling (IRB) is configured for a bridge domain or a VPLS routing instance, then you must configure a normalized VLAN using one of the following statements:
  - vlan-id *vlan-number*
  - vlan-id none
  - vlan-tags outer *outer-vlan-number* inner *inner-vlan-number*
- Use the vlan-id all statement to configure bridging for several VLANS with minimal amount of configuration and switch resources. See “One VPLS Instance for Several VLANs” on page 41 for an example of this configuration.

Chapter 4

MX-series Examples Using VLANs and VPLS

This chapter provides configuration examples to help you effectively configure a network of MX-series routers for a bridge domain or virtual private LAN service (VPLS) environment. The emphasis here is on choosing normalized virtual LAN (VLAN) configurations. The VPLS configuration is not covered in this chapter. For more information about configuring Ethernet pseudowires as part of VPLS, see the *JUNOS Feature Guide*.

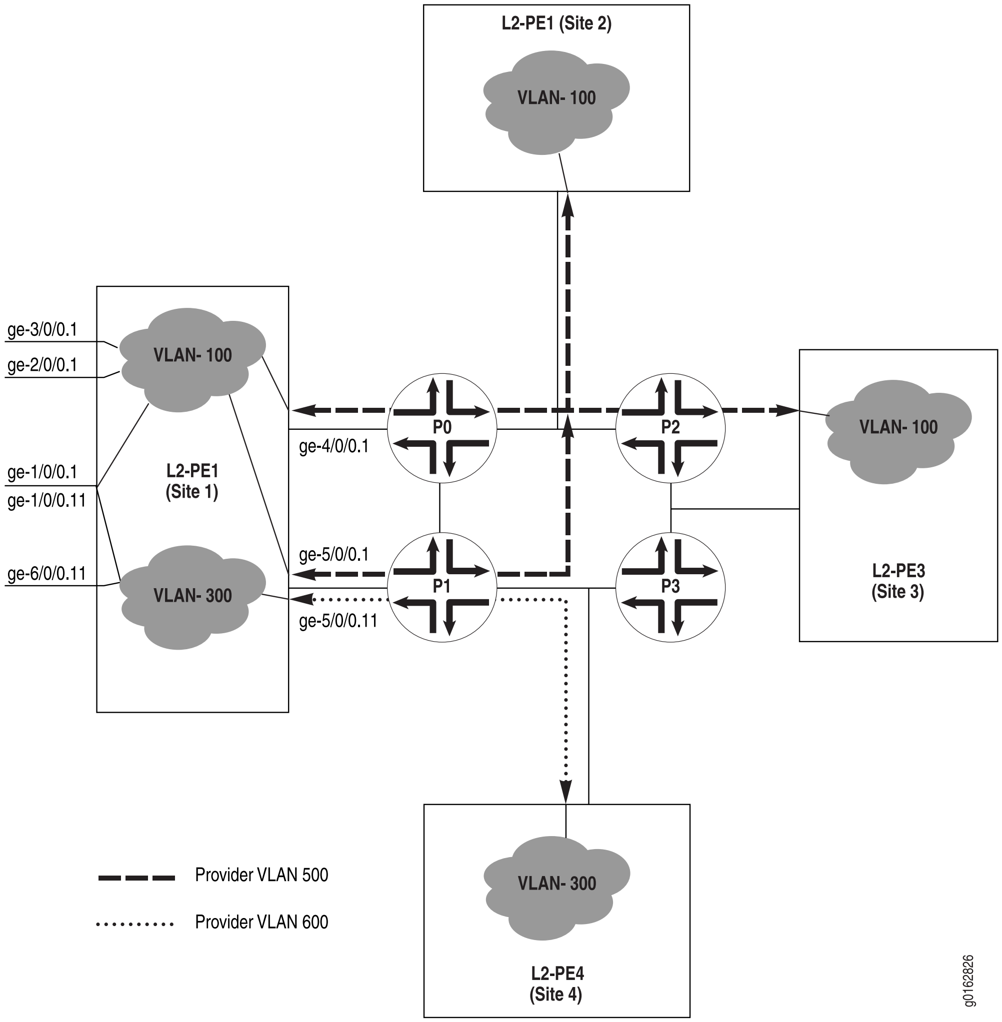
NOTE: This chapter does not present exhaustive configuration listings for all routers in the figures. However, you can use it with a broader configuration strategy to complete the MX-series router network configurations.

This chapter discusses the following topics:

Provider Bridge Network with Normalized VLAN Tags

Consider the provider bridge network shown in Figure 7 on page 34.

Figure 7: Provider Bridge Network Using Normalized VLAN Tags



The Layer 2 (L2) provider edge (PE) routers are MX-series routers. Each site is connected to two provider (P) routers for redundancy, although both links are only shown for L2-PE1 at Site 1. Site 1 is connected to P0 and P1 (as shown), Site 2 is connected to P0 and P2 (not shown), Site 3 is connected to P2 and P3 (as shown), and Site 4 is connected to P1 and P3 (as shown). VPLS pseudowires configured on the PE and P routers carry traffic between the sites.

The VLANs' bridging paths are shown with distinct dashed and dotted lines. The VLANs at each site are:

- L2-PE1 at Site 1: VLAN 100 and VLAN 300
- L2-PE2 at Site 2: VLAN 100
- L2-PE3 at Site 3: VLAN 100
- L2-PE4 at Site 4: VLAN 300

NOTE: The configurations in this chapter are only partial examples of complete and functional router configurations. Do not copy these configurations and use them directly on an actual system.

The following is the configuration of interfaces, virtual switches, and bridge domains for MX-series router L2-PE1:

```
[edit]
interfaces ge-1/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
   }
   unit 11 {
      encapsulation vlan-bridge;
      vlan-id 301;
   }
}
interface ge-2/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
   }
}
interface ge-3/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-bridge;
      vlan-id 200; # NOTE: 200 is translated to normalized VLAN vlaue
   }
}
interfaces ge-4/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-bridge;
      vlan-tags outer 500 inner 100; # This places two VLAN tags on the provider
                                     # pseudowire
   }
}
interfaces ge-5/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-bridge;
      vlan-tags outer 500 inner 100; # This places two VLAN tags on the provider
                                     # pseudowire
   }
   unit 11 {
```

```
            encapsulation vlan-bridge;
            vlan-tags outer 600 inner 300; # This places two VLAN tags on the provider
                                                    # pseudowire
        }
    }
    interfaces ge-6/0/0 {
        encapsulation flexible-ethernet-services;
        flexible-vlan-tagging;
        unit 11 {
            encapsulation vlan-bridge;
            vlan-id 300;
        }
    }
    routing-instances {
        customer-c1-virtual-switch {
            instance-type virtual-switch ;
            bridge-domains {
                c1-vlan-100 {
                    domain-type bridge;
                    vlan-id 100; # Customer VLAN 100 uses these five logical interfaces
                    interface ge-1/0/0.1;
                    interface ge-2/0/0.1;
                    interface ge-3/0/0.1;
                    interface ge-4/0/0.1;
                    interface ge-5/0/0.1;
                } # End of c1-vlan-100
            } # End of bridge-domains
        } # End of customer-c1-virtual-switch
        customer-c2-virtual-switch {
            instance-type virtual-switch ;
            bridge-domains {
                c2-vlan-300 {
                    domain-type bridge;
                    vlan-id 300; # Customer VLAN 300 uses these three logical interfaces
                    interface ge-1/0/0.11;
                    interface ge-5/0/0.11;
                    interface ge-6/0/0.11;
                } # End of c1-vlan-300
            } # End of bridge-domains
        } # End of customer-c2-virtual-switch
    } # end of routing-instances
```

Bridge domain c1–vlan-100 for customer-c1–virtual-switch has five logical interfaces:

- Logical interface ge-1/0/0.1 configured on physical port ge-1/0/0.
- Logical interface ge-2/0/0.1 configured on physical port ge-2/0/0.
- Logical interface ge-3/0/0.1 configured on physical port ge-3/0/0.
- Logical interface ge-4/0/0.1 can exist on an extended port/subinterface defined by the pair ge-4/0/0 and outer-vlan-tag 500.
- Logical interface ge-5/0/0.1 can exist on an extended port/subinterface defined by the pair ge-5/0/0 and outer-vlan-tag 500.

The association of the received packet to a logical interface is done by matching the VLAN tags of the received packet with the VLAN tags configured on one of the logical interfaces on that physical port. The vlan-id 100 configuration within the bridge domain c1–vlan-100 sets the normalized VLAN value to 100.

The following happens as a result of this configuration:

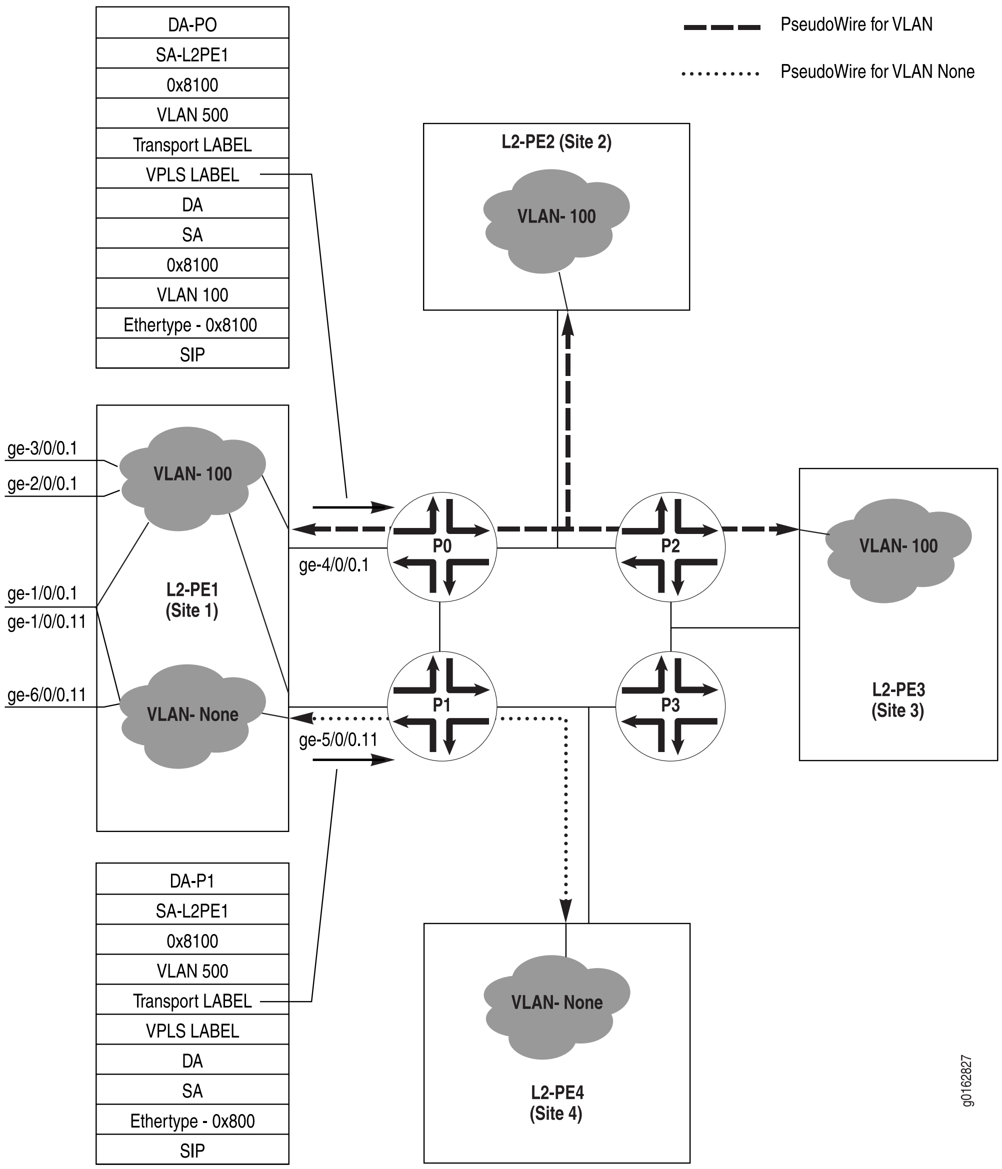
- Packets received on logical interfaces ge-1/0/0.1 or ge-2/0/0.1 with a single VLAN tag of 100 in the frame are accepted.
- Packets received on logical interface ge-3/0/0.1 with a single VLAN tag of 200 in the frame are accepted and have their tag values translated to the normalized VLAN tag value of 100.
- Packets received on logical interfaces ge-4/0/0.1 and ge-5/0/0.1 with outer tag values of 500 and inner tag values of 100 are accepted.
- Unknown source MAC addresses and unknown destination MAC addresses are learned based on their normalized VLAN values of 100 or 300.
- All packets sent on a logical interface always have their associated vlan-id value(s) in their VLAN tag fields.

Configuration and function of bridge domain c2-vlan-300 for customer-c2-virtual-switch is similar to, but not identical to, that of bridge domain c1-vlan-100 for customer-c1-virtual-switch.

VPLS Labels and VLAN Tags

Consider the VPLS network shown in Figure 8 on page 38.

Figure 8: VLAN Tags and VPLS Labels



The L2 PE routers are MX-series routers. Each site is connected to two P routers for redundancy, although both links are only shown for L2-PE1 at Site 1. Site 1 is connected to P0 and P1, Site 2 is connected to P0 and P2 (not shown), Site 3 is connected to P2 and P3, and Site 4 is connected to P1 and P3. VPLS pseudowires configured on the PE and P routers carry traffic between the sites.

The pseudowires for the VPLS instances are shown with distinct dashed and dotted lines. The VLANs at each site are:

- L2-PE1 at Site 1: VLAN 100 and VLAN 300
- L2-PE2 at Site 2: VLAN 100
- L2-PE3 at Site 3: VLAN 100
- L2-PE4 at Site 4: VLAN 300

Service provider SP-1 is providing VPLS services for customer C1 and C2. L2-PE1 is configured with a VPLS instance called customer-c1-vsi. The VPLS instance sets up pseudowires to remote Site 2 and Site 3. L2-PE1 is also configured with a VPLS instance called customer-c2-vsi. The VPLS instance sets up a pseudowire to remote Site 4.

The following is the configuration of interfaces, virtual switches, and bridge domains for MX-series router L2-PE1:

```
[edit]
interfaces ge-1/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id 100;
   }
   unit 11 {
      encapsulation vlan-vpls;
      vlan-id 301;
   }
}
interfaces ge-2/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id 100;
   }
}
interfaces ge-3/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id 200; # Should be translated to normalized VLAN value
   }
}
interfaces ge-6/0/0 {
   encapsulation flexible-ethernet-services;
   flexible-vlan-tagging;
   unit 11 {
      encapsulation vlan-vpls;
      vlan-id 302;
   }
}
```

```
routing-instances {
    customer-c1-vsi {
        instance-type vpls;
        vlan-id 100;
        interface ge-1/0/0.1;
        interface ge-2/0/0.1;
        interface ge-3/0/0.1;
    } # End of customer-c1-vsi
    customer-c2-vsi {
        instance-type vpls;
        vlan-id none; # This will remove the VLAN tags from packets sent on VPLS for
            customer 2
        interface ge-1/0/0.11;
        interface ge-6/0/0.11;
    } # End of customer-c2-vsi
} # End of routing-instances
```

Consider the first VLAN for customer C1. The vlan-id 100 statement in the VPLS instance called customer-c1-vsi sets the normalized VLAN to 100. All packets sent over the pseudowires have a VLAN tag of 100.

The following happens on VLAN 100 as a result of this configuration:

- Packets received on logical interfaces ge-1/0/0.1 or ge-2/0/0.1 with a single VLAN tag of 100 in the frame are accepted.
- Packets received on logical interface ge-3/0/0.1 with a single VLAN tag of 200 in the frame are accepted and have their tag values translated to the normalized VLAN tag value of 100.
- Unknown source MAC addresses and unknown destination MAC addresses are learned based on their normalized VLAN values of 100.
- All packets sent on the VPLS pseudowire have vlan-id 100 in their VLAN tag fields.

Now consider the second VLAN for Customer C2. The vlan-id none statement in the VPLS instance called customer-c2-vsi removes the incoming VLAN tags before the packets are sent over the VPLS pseudowires.

The following happens on the C2 VLAN as a result of the vlan-id none configuration:

- A MAC table is created for each instance of vlan-id none. All MAC addresses learned over the interfaces belonging to this VPLS instance are added to this table. The received or configured VLAN tags are not considered when the MAC addresses are added to this table. This is a case of shared VLAN learning.
- Packets with a single VLAN tag value of 301 are accepted on interface ge-1/0/0.11. The VLAN tag value 301 is then popped and removed from the frame of this packet.
- Packets with a single VLAN tag value of 302 are accepted on interface ge-6/0/0.11. The VLAN tag value 302 is then popped and removed from the frame of this packet.
- All packets sent on pseudowires will not have any VLAN tags used to identify the incoming Layer 2 logical interface.

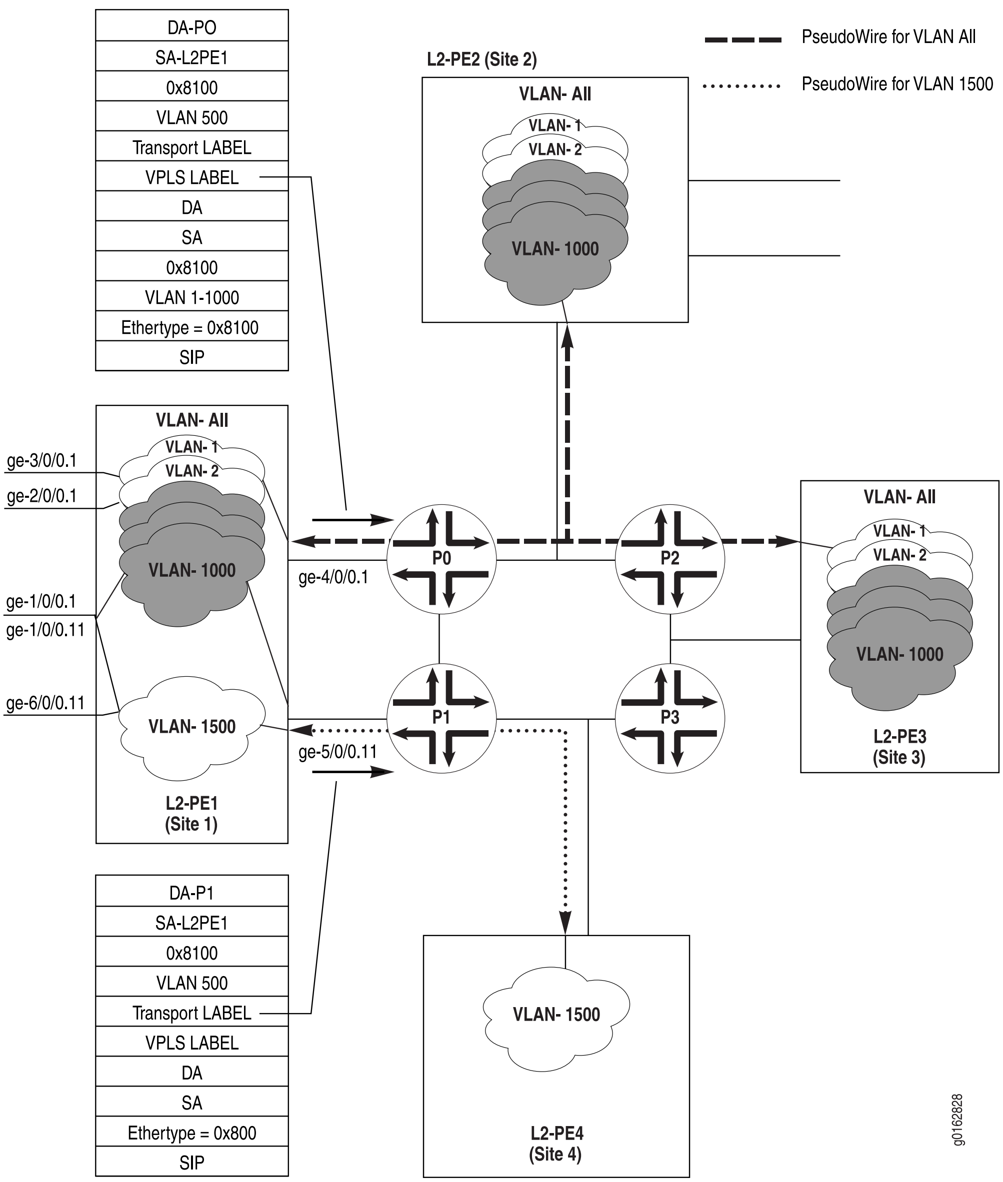
NOTE: The packet can still contain other customer VLAN tags.

- Packets received from pseudowires are looked up in the MAC table associated with the VPLS instance. Any customer VLAN tags in the frame are ignored.

One VPLS Instance for Several VLANs

Consider the VPLS network shown in Figure 9 on page 41.

Figure 9: Many VLANs on one VPLS Instance



The L2 PE routers are MX-series routers. Each site is connected to two P routers for redundancy, although both links are only shown for L2-PE1 at Site 1. Site 1 is connected to P0 and P1, Site 2 is connected to P0 and P2 (not shown), Site 3 is connected to P2 and P3, and Site 4 is connected to P1 and P3. VPLS pseudowires configured on the PE and P routers carry traffic between the sites.

The pseudowires for the VPLS instances are shown with distinct dashed and dotted lines. Most sites have multiple VLANs configured.

Service provider SP-1 is providing VPLS services for customer C1, services that could span several sites. Now customer C1 can have many VLANs in the range from 1 through 1000 (for example).

If VLANs 1 through 1000 for customer C1 span the same sites, then the vlan-id all and vlan-range statements provide a way to switch all of these VLANs with a minimum configuration effort and fewer switch resources.

NOTE: You cannot use the vlan-id all statement if you configure an IRB interface on one or more of the VLANs.

The following example illustrates the use of the vlan-id all statement:

```
[edit]
interfaces ge-1/0/0 {
   flexible-vlan-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id-range 1-1000;
   }
   unit 11 {
      encapsulation vlan-vpls;
      vlan-id 1500;
   }
}
interfaces ge-2/0/0 {
   flexible-vlan-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id-range 1-1000; # Note the use of the VLAN id range statement.
   }
}
interfaces ge-3/0/0/ {
   flexible-vlan-services;
   flexible-vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id 1-1000;
   }
}
interfaces ge-6/0/0 {
   flexible-vlan-services;
```

```
    flexible-vlan-tagging;
    unit 11 {
      encapsulation vlan-vpls;
      vlan-id 1500;
    }
  }
  routing-instances {
    customer-c1-v1-to-v1000 {
      instance-type vpls;
      vlan-id all; # Note the use of the VLAN id all statement
      interface ge-1/0/0.1;
      interface ge-2/0/0.1;
      interface ge-3/0/0.1;
    } # End of customer-c1-v1-to-v1000
    customer-c1-v1500 {
      instance-type vpls;
      vlan-id 1500;
      interface ge-1/0/0.11;
      interface ge-6/0/0.11;
    } # End of customer-c1-v1500
  } # End of routing-instances
```

Note the use of the vlan-id all and vlan-id-range statements in the VPLS instance called customer-c1-v1-to-v1000. The vlan-id all statement implicitly creates multiple learning domains, each with its own normalized VLAN.

The following happens as a result of the vlan-id all configuration:

- Packets received on logical interfaces ge-1/0/0.1 , or ge-2/0/0.1, or ge-3/0/0.1, with a single VLAN tag in the range from 1 through 1000 in the frame are accepted.
- Unknown source MAC addresses and unknown destination MAC addresses are learned based on their normalized VLAN values of 1 through 1000.
- All packets sent on the VPLS pseudowire have a normalized VLAN tag after the source MAC address field in the encapsulated Ethernet packet.
- Although there are only three logical interfaces in the VPLS instance called customer-c1-v1-to-v1000, the same MAC address (for example, M1) can be learned on different logical interfaces for different VLANs. For example, MAC address M1 could be learned on logical interface ge-1/0/0.1 for VLAN 500 and also on logical interface ge-2/0/0.1 for VLAN 600.

Chapter 5
Configuring Ethernet OAM

This chapter provides configuration examples to help you effectively configure Ethernet Operation, Administration, and Maintenance (OAM) on a network of MX-series routers. For more information about configuring OAM parameters on Ethernet interfaces, see the *JUNOS Interfaces Configuration Guide*.

NOTE: This chapter does not present exhaustive configuration listings for all routers in the figures. However, you can use it with a broader configuration strategy to complete the MX-series router network Ethernet OAM configurations.

This chapter discusses the following topics:

Overview of Ethernet OAM

Ethernet OAM provides the tools that network management software and network managers can use to determine how a network of Ethernet links is functioning. OAM can provide simple link-level information, provide performance statistics, or track end-to-end connectivity across the network. Simple link fault management (LFM) for Ethernet links is defined in IEEE 802.3ah. The most complete connectivity fault management (CFM) is defined in IEEE 802.1ag. This chapter emphasizes the use of CFM in a Metro Ethernet environment.

CFM can be used to monitor an Ethernet network at a per-service level, unlike LFM, which functions at the physical link level. The service monitored could be a virtual local area network (VLAN), concatenation of VLANs or a virtual private LAN service (VPLS) instance.

The major features of CFM are:

- Fault monitoring using the continuity check protocol
- Path discovery and fault verification using the linktrace protocol

- Fault isolation using the loopback protocol

CFM partitions the service network into various administrative domains. For example, operators, providers, and customers may be part of different administrative domains. Each administrative domain is mapped into one maintenance domain providing enough information to perform its own management, thus avoiding security breaches and making end-to-end monitoring possible. Each maintenance domain is associated with a level. Level allocation is based on the network hierarchy, where outermost domains are assigned a higher level than the innermost domains. In a CFM maintenance domain, each service instance is called a maintenance association. A maintenance association can be thought as a full mesh of maintenance endpoints (MEPs) having similar characteristics. MEPs are active CFM entities generating and responding to CFM protocol messages. The following examples below use CFM to monitor connectivity over a VPLS and bridge network.

In all the examples in this chapter, CFM can be used at two levels:

- By the service provider to check the connectivity among its provider edge (PE) routers
- By the customer to check the connectivity among its customer edge (CE) routers

NOTE: The configured customer CFM level must be greater than service provider CFM level.

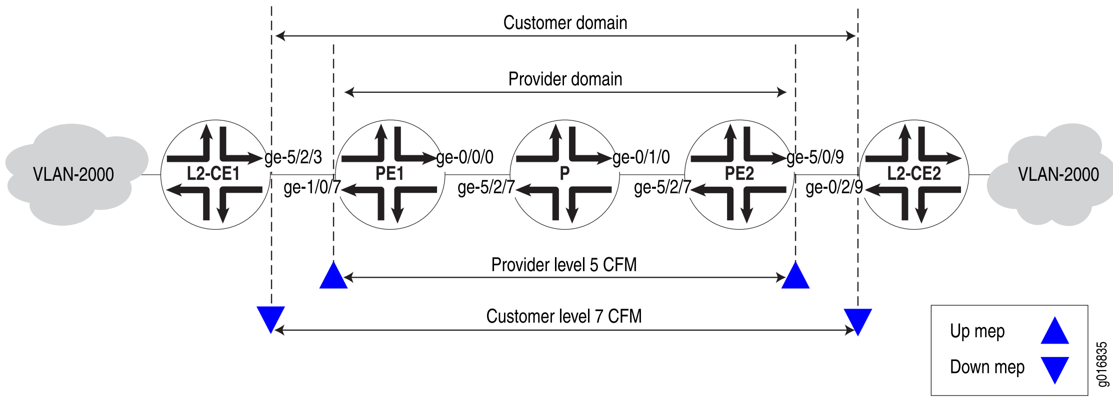
NOTE: The configurations in this chapter are only partial examples of complete and functional router configurations. Do not copy these configurations and use them directly on an actual system.

Ethernet CFM over VPLS

In this example, both the customer and service provider are running Ethernet CFM over a VPLS and a multiprotocol label switching (MPLS) network. The network is shown in Figure 10 on page 47. The customer has configured Ethernet CFM on MX-series routers L2-CE1 and L2-CE2. The service provider has configured Ethernet CFM on MX-series routers PE1, P, and PE2.

The service provider is using CFM level 5 and the customer is using CFM level 7. The boundaries are marked with “up mep” and “down mep” CFM terminology in the figure.

Figure 10: Ethernet OAM with VPLS



The following are the configurations of the VPLS and CFM on the service provider routers.

Configuration of PE1

```
[edit chassis]
fpc 5 {
   pic 0 {
      tunnel-services {
         bandwidth 1g;
      }
   }
}


[edit interfaces]
ge-1/0/7 {
   encapsulation flexible-ethernet-services;
   vlan-tagging;
   unit 1 {
      encapsulation vlan-vpls;
      vlan-id 2000;
   }
}
ge-0/0/0 {
   unit 0 {
      family inet {
         address 10.200.1.1/24;
      }
      family mpls;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 10.255.168.231/32 {
            primary;
         }
         address 127.0.0.1/32;
      }
   }
}
```

```
[edit routing-instances]
vpls-vlan2000 {
   instance-type vpls;
   vlan-id 2000;
   interface ge-1/0/7.1;
   route-distinguisher 10.255.168.231:2000;
   vrf-target target:1000:1;
   protocols {
      vpls {
         site-range 10;
         site vlan2000-PE1 {
            site-identifier 2;
         }
      }
   }
}

[edit protocols]
rsvp {
   interface ge-0/0/0.0;
}
mpls {
   label-switched-path PE1-to-PE2 {
      to 10.100.1.1;
   }
   interface ge-0/0/0.0;
}
bgp {
   group PE1-to-PE2 {
      type internal;
      local-address 10.200.1.1;
      family l2vpn {
         signaling;
      }
      local-as 65000;
      neighbor 10.100.1.1;
   }
}
ospf {
   traffic-engineering;
   reference-bandwidth 4g;
   area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface ge-0/0/0.0;
   }
}
oam {
   ethernet {
      connectivity-fault-management {
         maintenance-domain customer-site1 {
            level 5;
            maintenance-association customer-site1 {
```

```
                    continuity-check {
                        interval 1s;
                    }
                    mep 100 {
                        interface ge-1/0/7.1;
                        direction up;
                        auto-discovery;
                    }
                }
            }
        }
    }
}
```

Configuration of PE2

```
[edit chassis]
fpc 5 {
    pic 0 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}

[edit interfaces]
ge-5/0/9 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
        encapsulation vlan-vpls;
        vlan-id 2000;
    }
}
ge-5/2/7 {
    unit 0 {
        family inet {
            address 10.100.1.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.168.230/32 {
                primary;
            }
            address 127.0.0.1/32;
        }
    }
}

[edit routing-instances]
vpls-vlan2000 {
    instance-type vpls;
    vlan-id 2000;
```

```
    interface ge-5/0/9.1;
    route-distinguisher 10.255.168.230:2000;
    vrf-target target:1000:1;
    protocols {
        vpls {
            site-range 10;
            site vlan2000-PE2 {
                site-identifier 1;
            }
        }
    }
}

[edit protocols]
rsvp {
    interface ge-5/2/7.0;
}
mpls {
    label-switched-path PE2-to-PE1 {
        to 10.200.1.1;
    }
    interface ge-5/2/7.0;
}
bgp {
    group PE2-to-PE1 {
        type internal;
        local-address 10.100.1.1;
        family l2vpn {
            signaling;
        }
        local-as 65000;
        neighbor 10.200.1.1;
    }
}
ospf {
    traffic-engineering;
    reference-bandwidth 4g;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface ge-5/2/7.0;
    }
}
oam {
    ethernet {
        connectivity-fault-management {
            maintenance-domain customer-site1 {
                level 5;
                maintenance-association customer-site1 {
                    continuity-check {
                        interval 1s;
                    }
                    mep 200 {
```

```
                        interface ge-5/0/9.1;
                        direction up;
                        auto-discovery;
                    }
                }
            }
        }
    }
}
```

Configuration of P router

MPLS only, no CFM needed:

```
[edit]
interfaces {
    ge-5/2/7 {
        # Connected to PE1
        unit 0 {
            family inet {
                address 10.200.1.10/24;
            }
            family mpls;
        }
    }
    ge-0/1/0 {
        # Connected to PE2
        unit 0 {
            family inet {
                address 10.100.1.10/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0{
            family inet {
                address 10.255.168.240/32;
            }
        }
    }
}

[edit]
protocols {
    rsvp {
        interface ge-0/1/0.0;
        interface ge-5/2/7.0;
    }
    mpls {
        interface ge-0/1/0.0;
        interface ge-5/2/7.0;
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface all;
```

```
            interface fxp0.0 {
                disable;
            }
            interface ge-0/1/0.0;
            interface ge-5/2/7.0;
        }
    }
}
```

CFM on L2-CE1 Here is the configuration of CFM on L2-E1:

```
[edit interfaces]
ge-5/2/3 {
    vlan-tagging;
    unit 0 {
        vlan-id 2000;
    }
}

[edit protocols oam]
ethernet {
    connectivity-fault-management {
        maintenance-domain customer {
            level 7;
            maintenance-association customer-site1 {
                continuity-check {
                    interval 1s;
                }
                mep 800 {
                    interface ge-5/2/3.0;
                    direction down;
                    auto-discovery;
                }
            }
        }
    }
}
```

CFM on L2-CE2 Here is the configuration of CFM L2-CE2:

```
[edit interfaces]
ge-0/2/9 {
    vlan-tagging;
    unit 0 {
        vlan-id 2000;
    }
}

[edit protocols oam]
ethernet {
    connectivity-fault-management {
        maintenance-domain customer {
            level 7;
            maintenance-association customer-site1 {
                continuity-check {
```

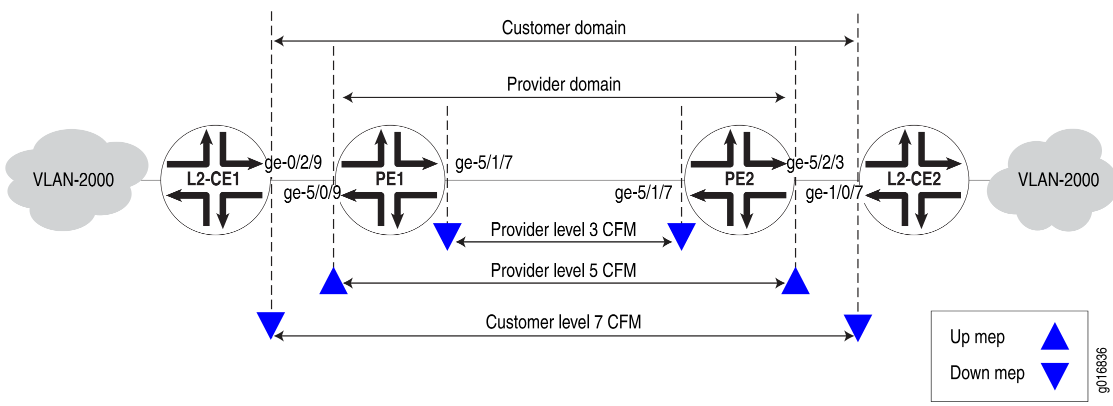
```
                interval 1s;
            }
            mep 700 {
                interval ge-0/2/9.0;
                direction down;
                auto-discovery;
            }
        }
    }
}
}
```

Ethernet CFM on Bridge Connections

In this example, both the customer and service provider are running Ethernet CFM over a simple bridge network. The network is shown in Figure 11 on page 53. The customer has configured Ethernet CFM on MX-series routers L2-CE1 and L2-CE2. The service provider has configured Ethernet CFM on MX-series routers PE1 and PE2.

The service provider is using CFM level 3 for the link between PE1 and PE2 and level 5 from one CE facing port to the other. The customer is using CFM level 7. The boundaries are marked with "up mep" and "down mep" CFM terminology in the figure.

Figure 11: Ethernet CFM over a Bridge Network



Here are the configurations of CFM on the customer routers.

CFM on L2-CE1

```
[edit interfaces]
ge-0/2/9 {
    vlan-tagging;
    unit 0 {
        vlan-id 2000;
    }
}

[edit protoccols oam ethernet]
connectivity-fault-management {
```

```
    maintenance-domain customer {
        level 7;
        maintenance-association customer-site1 {
            continuity-check {
                interval 1s;
            }
            mep 700 {
                interval ge-0/2/9.0;
                direction down;
                auto-discovery;
            }
        }
    }
}
```

CFM on L2-CE2

```
[edit interfaces]
ge-1/0/7 {
    vlan-tagging;
    unit 0 {
        vlan-id 2000;
    }
}

[edit protoccols oam ethernet]
connectivity-fault-management {
    maintenance-domain customer {
        level 7;
        maintenance-association customer-site2 {
            continuity-check {
                interval 1s;
            }
            mep 800 {
                interface ge-1/0/7.0;
                direction down;
                auto-discovery;
            }
        }
    }
}
```

Here are the configurations of CFM on the provider routers.

CFM on PE1

```
[edit interfaces]
ge-5/0/9 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id 2000;
    }
}
ge-5/1/7 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
```

```
        encapsulation vlan-bridge;
        vlan-id 2000;
    }
}

[edit bridge-domains]
bridge-vlan2000 {
    domain-type bridge;
    vlan-id 2000;
    interface ge-5/0/9.0;
    interface ge-5/1/7.0;
}

[edit protocols oam ethernet connectivity-fault-management]
maintenance-domain provider-outer {
    level 5;
    maintenance-association provider-outer-site1 {
        continuity-check {
            interval 1s;
        }
        mep 200 {
            interface ge-5/0/9.0;
            direction up;
            auto-discovery;
        }
    }
}
maintenance-domain provider-inner {
    level 3;
    maintenance-association provider-inner-site1 {
        continuity-check {
            interval 1s;
        }
        mep 200 {
            interface ge-5/1/7.0;
            direction down;
            auto-discovery;
        }
    }
}
```

CFM on PE2

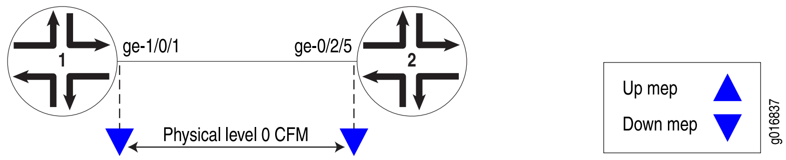
```
[edit interfaces]
ge-5/1/7 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id 2000;
    }
}
ge-5/2/3 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
```

```
        vlan-id 2000;
    }
}

[edit bridge-domains]
bridge-vlan2000 {
    domain-type bridge;
    interface ge-5/2/3.0;
    interface ge-5/1/7.0;
}

[edit protocols oam ethernet connectivity-fault-management]
maintenance-domain provider-outer {
    level 5;
    maintenance-association provider-outer-site1 {
        continuity-check {
            interval 1s;
        }
        mep 100 {
            interface ge-5/2/3.0;
            direction up;
            auto-discovery;
        }
    }
}
maintenance-domain provider-inner {
    level 3;
    maintenance-association provider-inner-site1 {
        continuity-check {
            interval 1s;
        }
        mep 100 {
            interface ge-5/1/7.0;
            direction down;
            auto-discovery;
        }
    }
}
```

Ethernet CFM on Physical Interfaces

CFM can be used to monitor the physical link between two routers. This functionality is similar to that supported by theIEEE 802.3ah LFM protocol. In the following examples, two routers (Router #1 and Router #2) are connected by a point-to-point Gigabit Ethernet link. The link between these two routers is monitored using CFM. This is shown in Figure 12 on page 57. The single boundary is a “down mep” in CFM terminology.

Figure 12: Ethernet CFM on Physical Interfaces



Router 1 Configure the interface and CFM:

```
[edit]
interfaces ge-1/0/1 {
   unit 0 {
      family inet;
   }
}

protocols {
   oam {
      ethernet {
         connectivity-fault-management {
            maintenance-domain private {
               level 0;
               maintenance-association private-ma {
                  continuity-check {
                     interval 1s;
                  }
                  mep 100 {
                     interface ge-1/0/1;
                     direction down;
                     auto-discovery;
                  }
               }
            }
         }
      }
   }
}
```

The configuration on Router 2 mirrors that on Router 1.

Router 2 Configure the interface and CFM:

```
[edit]
interfaces ge-0/2/5 {
   unit 0 {
      family inet;
   }
}

protocols {
   oam {
      ethernet {
         connectivity-fault-management {
            maintenance-domain private {
               level 0;
```

```
                    maintenance-association private-ma {
                        continuity-check {
                            interval 1s;
                        }
                        mep 100 {
                            interface ge-0/2/5;
                            direction down;
                            auto-discovery;
                        }
                    }
                }
            }
        }
    }
}
```

Ethernet LFM

LFM can be used for physical link-level fault detection and management. The IEEE 802.3ah LFM works across a point-to-point Ethernet link either directly connected or through repeaters.

LFM provides the following functions:

- Failure detection on physical links in both directions, as well as unidirectional failures.
- Ability to put a port in link-loopback mode remotely for diagnostics.
- Report and receive link error events such as framing or symbol errors.

LFM runs at the physical or aggregated interface level. When configured on an aggregated interface, LFM is run individually on each member link. LFM is a link-layer protocol and does not need a Layer 3 (IPv4 or IPv6) address to operate. This allows for LFM to function on circuit cross-connect/transport cross-connect (CCC/TCC) encapsulated interfaces.

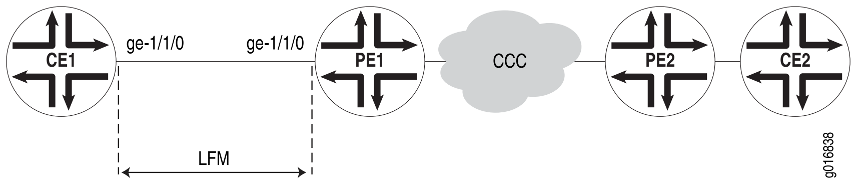
The following examples show how LFM is configured in various situations:

- Ethernet LFM Between PE and CE on page 58
- Ethernet LFM for CCC on page 60
- Ethernet LFM for Aggregated Ethernet on page 61
- Ethernet LFM with Loopback Support on page 62

Ethernet LFM Between PE and CE

In this example, LFM is enabled on an IP link between the provider edge (PE) and customer edge (CE) interfaces. If the link goes down, the fault will be detected by LFM and the interfaces on both sides will be marked Link-Layer-Down. This results in notifications to various subsystems (for example, routing) which will take appropriate action. The link running LFM is shown in Figure 13 on page 59.

Figure 13: Ethernet LFM Between PE and CE



PE Router Configure LFM on the PE router:

```
[edit]
interfaces ge-1/1/0 {
   unit 0 {
      family inet {
         address 11.11.11.1/24;
      }
   }
}

protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ge-1/1/0 {
               pdu-interval 1000;
               pdu-threshold 5;
            }
         }
      }
   }
}
```

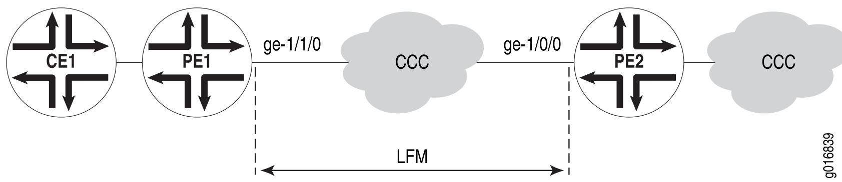
CE Router Configure LFM on the CE router:

```
[edit]
interfaces ge-1/1/0 {
   unit 0 {
      family inet {
         address 11.11.11.2/24;
      }
   }
}

protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ge-1/1/0 {
               pdu-interval 1000;
               pdu-threshold 5;
            }
         }
      }
   }
}
```

Ethernet LFM for CCC

In this example, LFM is configured between two PEs (PE1 and PE2) connected using CCC. With LFM in place, a link fault will be detected immediately, instead of depending on routing protocols to find the fault on end-to-end CCC connection. This also helps in detecting the exact failed link instead of only finding that the end-to-end CCC connectivity has failed. Also, because LFM runs at the link-layer level, it does not need a IP address to operate and so can be used where bidirectional fault detection (BFD) cannot. The links running LFM are shown in Figure 14 on page 60.

Figure 14: Ethernet LFM for CCC



PE1 Router Configure LFM on the PE1 router with CCC:

```
[edit]
interfaces ge-1/1/0 {
   encapsulation ethernet-ccc;
   unit 0;
}

protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ge-1/1/0 {
               pdu-interval 1000;
               pdu-threshold 5;
            }
         }
      }
   }
}
```

PE2 Router Configure LFM on the PE2 router with CCC:

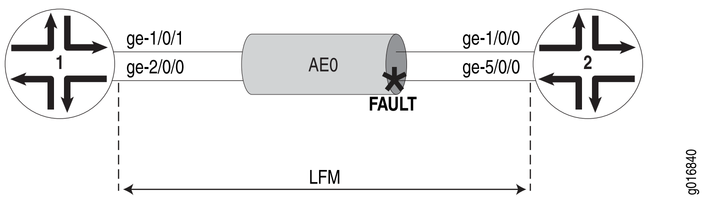
```
[edit]
interfaces ge-1/0/0 {
   encapsulation ethernet-ccc;
   unit 0;
}

protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ge-1/0/0 {
               pdu-interval 1000;
               pdu-threshold 5;
            }
```

```
            }
         }
      }
   }
```

Ethernet LFM for Aggregated Ethernet

In this example, LFM is configured on an aggregated Ethernet interface (AE0) between Router #1 and Router #2. When configured on aggregated Ethernet, LFM runs on all the individual member links. LFM is enabled or disabled on the member links as they are added or deleted from the aggregation group. The status of individual links is used to determine the status of the aggregated interface. The use of LFM with aggregated Ethernet is shown in Figure 15 on page 61.

Figure 15: Ethernet LFM for Aggregated Ethernet



Router #1 Configure LFM on Router #1 for AE0:

```
[edit]
chassis {
   aggregated-devices {
      ethernet {
         device-count 1;
      }
   }
}
interfaces ge-1/0/1 {
   gigether-options {
      802.3ad ae0;
   }
}
interfaces ge-2/0/0 {
   gigether-options {
      802.3ad ae0;
   }
}
interfaces ae0 {
   unit 0 {
      family inet {
         address 11.11.11.2/24;
      }
   }
}
protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ae0;
```

```
            }
         }
      }
   }
```

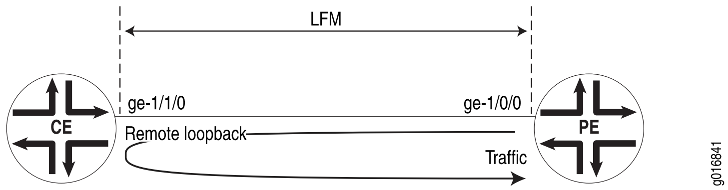
Router #2 Configure LFM on Router #2 for AE0:

```
[edit]
chassis {
   aggregated-devices {
      ethernet {
         device-count 1;
      }
   }
}
interfaces ge-1/0/0 {
   gigether-options {
      802.3ad ae0;
   }
}
interfaces ge-5/0/0 {
   gigether-options {
      802.3ad ae0;
   }
}
interfaces ae0 {
   unit 0 {
      family inet {
         address 11.11.11.1/24;
      }
   }
}
protocols {
   oam {
      ethernet {
         link-fault-management {
            interface ae0;
         }
      }
   }
}
```

Ethernet LFM with Loopback Support

In this example, LFM is configured between PE and CE. The PE can put the CE in remote loopback mode. This allows the PE to have all the traffic sent to the CE looped back for diagnostics purposes, as shown in Figure 16 on page 63.

Figure 16: Ethernet LFM with Loopback Support



PE Router Configure LFM loopback on the PE router:

```
[edit]
interfaces ge-1/0/0 {
  unit 0 {
    family inet {
      address 11.11.11.1/24;
    }
  }
}
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface ge-1/0/0 {
          pdu-interval 1000;
          pdu-threshold 5;
          remote-loopback;
        }
      }
    }
  }
}
```

CE Router Configure LFM loopback on the CE router:

```
[edit]
interfaces ge-1/1/0 {
  unit 0 {
    family inet {
      address 11.11.11.2/24;
    }
  }
}
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface ge-1/1/0 {
          pdu-interval 1000;
          pdu-threshold 5;
          negotiation-options {
            allow-remote-loopback;
          }
        }
      }
    }
  }
```

```
}
```

Chapter 6

Configuring MX-series Filters

MX-series routers support firewall filters for the bridge and vpls protocol families. You configure these firewall filters to control traffic within bridge domains and VPLS instances. This chapter explores some of the ways that filters can be used in an Layer 2 (L2) environment to control traffic.

MX-series firewall filters can be applied to:

- Input interfaces
- Output interfaces
- Input to the L2 forwarding table

You use a firewall filter after taking the following two steps:

1. You configure any policers and the firewall filter at the [edit firewall] hierarchy level.
2. You apply the properly configured firewall filter to an interface.

NOTE: You should deploy firewall filters carefully because it is easy to cause unforeseen side effects on all traffic, especially traffic that is not the intended target of the filter. For more information about configuring firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

This chapter provides the following information about JUNOS software firewall filters applied to MX-series routers at L2:

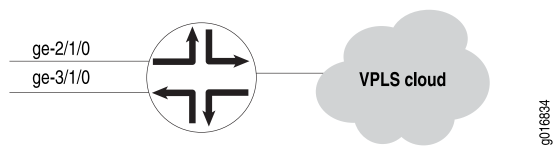
NOTE: This chapter does not present exhaustive configuration listings for all routers in the figures. However, you can use it with a broader configuration strategy to complete the MX-series router network Ethernet Operations, Administration, and Maintenance (OAM) configurations.

Policing and Marking Traffic Entering a VPLS Core

This example firewall filter allows a service provider to limit the aggregate broadcast traffic entering the virtual private LAN service (VPLS) core. The broadcast, unknown unicast, and non-IP multicast traffic received from one of the service provider's customers on a logical interface has a policer applied. The service provider has also configured a two-rate, three-color policer to limit the customer's IP multicast traffic. For more information on the configuration of policers, see the *JUNOS Class of Service Configuration Guide*.

The position of the router is shown in Figure 17 on page 66.

Figure 17: Policing and Marking Traffic Entering a VPLS Core



There are four major parts to the configuration:

- The policer for broadcast, unknown unicast, and non-IP multicast traffic. This example marks the loss priority as high if this type of traffic exceeds 50 Kbps.
- The two-rate, three-color policer for IP multicast traffic. This example configures a committed information rate (CIR) of 4 Mbps, a committed burst size of 256 Kbytes, a peak information rate of 4.1 Mbps, and a peak burst size of 256 Kbytes (the same as the CIR).
- The filter that applies the two policers to VPLS.
- The application of the filter to the customer interface configuration as an input filter.

Firewall Policer

This policer is used to limit the aggregate broadcast, unknown unicast, and non-IP multicast to 50 kbps:

```
[edit firewall]
policer bcast-unknown-unicast-non-ip-mcast-policer {
  if-exceeding {
    bandwidth-limit 50k;
    burst-size-limit 150k;
  }
  then loss-priority high;
}
```

Three-Color Policer

This policer is used to limit the IP multicast traffic:

```
[edit firewall]
three-color-policer ip-multicast-traffic-policer {
  two-rate {
    color-blind;
    committed-information-rate 4m;
    committed-burst-size 256k;
    peak-information-rate 4100000;
    peak-burst-size 256k;
```

```
    }
}
```

Firewall Filter This uses the two policers to limit and mark customer traffic. The first term marks the IP mulitcast traffic based on destination MAC address, and the second term polices the broadcast, unknown unicast, and non-IP multicast traffic:

```
[edit firewall]
family vpls {
    filter customer-1 {
        term t0 {
            from {
                destination-mac-address {
                    01:00:5e:00:00:00/24;
                }
            }
            then {
                three-color-policer {
                    two-rate ip-multicast-traffic-policer;
                }
                forwarding-class expedited-forwarding;
            }
        }
        term t1 {
            from {
                traffic-type [ broadcast unknown-unicast multicast ];
            }
            then policer bcast-unknown-unicast-non-ip-mcast-policer;
        }
    }
}
```

Apply Filter to Customer Interface Apply filter as an input filter to ge-2/1/0:

```
[edit]
interfaces {
    ge-2/1/0 {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 5 {
            encapsulation vlan-vpls;
            vlan-id 9;
            family vpls {
                filter {
                    input customer-1;
                }
            }
        }
    }
}
```

Filtering Frames by MAC Address

This example firewall filter finds frames with a certain source MAC address (88:05:00:29:3c:de/48), then counts and silently discards them. For more information about configuring firewall filter match conditions, see the *JUNOS Policy Framework*

Configuration Guide. The filter is applied to the VLAN configured as vlan100200 as an input filter on Router 1.

Router 1 Configure the firewall filter:

```
[edit firewall]
family bridge {
  filter evil-mac-address {
    term one {
      from {
        source-mac-address 88:05:00:29:3c:de/48;
      }
      then {
        count evil-mac-address; # Counts frame with the bad source MAC address
        discard;
      }
      term two {
        then accept; # Make sure to accept other traffic
      }
    }
  }
```

Apply to Virtual Switch Apply as an input filter to vlan100200 on Router 1:

```
[edit routing-instances virtual-switch-R1-1]
bridge-domains {
  vlan100200 {
    domain-type bridge;
    forwarding-options {
      filter {
        input evil-mac-address;
      }
    }
  }
}
```

Part 3
Index

Index