



Junos[®] OS

MX Series 3D Universal Edge Routers Solutions Guide

Release

13.1R1



Published: 2013-02-13

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

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® OS MX Series 3D Universal Edge Routers Solutions Guide

Release 13.1R1

Copyright © 2013, Juniper Networks, Inc.

All rights reserved.

Revision History

February 2013—Junos OS 13.1R1

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Abbreviated Table of Contents

	About This Guide	xiii
Part 1	Overview	
Chapter 1	Overview of Ethernet Solutions	3
Part 2	Basic Solutions for MX Series Routers	
Chapter 2	Basic Layer 2 Features on MX Series Routers	21
Chapter 3	Virtual Switches	39
Chapter 4	VLANs Within Bridge Domain and VPLS Environments	43
Chapter 5	Bulk Administration of Layer 2 Features on MX Series Routers	63
Chapter 6	Dynamic Profiles for VLAN Interfaces and Protocols	67
Chapter 7	MX Series Router as a DHCP Relay Agent	79
Chapter 8	MX Series Router in an ATM Ethernet Interworking Function	83
Part 3	Ethernet Filtering, Monitoring, and Fault Management Solutions for MX Series Routers	
Chapter 9	Layer 2 Firewall Filters	101
Chapter 10	IEEE 802.1ag OAM Connectivity-Fault Management	109
Chapter 11	ITU-T Y.1731 Ethernet Frame Delay Measurements	125
Chapter 12	IEEE 802.3ah OAM Link-Fault Management	143
Chapter 13	Ethernet Ring Protection	151
Part 4	Index	
	Index	185

Table of Contents

	About This Guide	xiii
	Junos Documentation and Release Notes	xiii
	Objectives	xiv
	Audience	xiv
	Supported Routing Platforms	xv
	Using the Indexes	xv
	Using the Examples in This Manual	xv
	Merging a Full Example	xv
	Merging a Snippet	xvi
	Documentation Conventions	xvi
	Documentation Feedback	xviii
	Requesting Technical Support	xviii
	Self-Help Online Tools and Resources	xix
	Opening a Case with JTAC	xix
Part 1	Overview	
Chapter 1	Overview of Ethernet Solutions	3
	Ethernet Terms and Acronyms	3
	Networking and Internetworking with Bridges and Routers	6
	Network Addressing at Layer 2 and Layer 3	7
	Networking at Layer 2: Benefits of Ethernet Frames	9
	Networking at Layer 2: Challenges of Ethernet MAC Addresses	10
	Networking at Layer 2: Forwarding VLAN Tagged Frames	11
	Networking at Layer 2: Forwarding Dual-Tagged Frames	13
	Networking at Layer 2: Logical Interface Types	14
	A Metro Ethernet Network with MX Series Routers	15
	Layer 2 Networking Standards	17
Part 2	Basic Solutions for MX Series Routers	
Chapter 2	Basic Layer 2 Features on MX Series Routers	21
	Layer 2 Features for a Bridging Environment	21
	Example Roadmap: Configuring a Basic Bridge Domain Environment	22
	Example Topology	22
	Example Scenario	23
	Example Configuration Summary	24
	Example Step: Configuring Interfaces and VLAN Tags	24
	Example Step: Configuring Bridge Domains	30
	Example Step: Configuring Spanning Tree Protocols	32
	Example Step: Configuring Integrated Bridging and Routing	34

Chapter 3	Virtual Switches	39
	Layer 2 Features for a Switching Environment	39
	Configuring Virtual Switches as Separate Routing Instances	40
Chapter 4	VLANs Within Bridge Domain and VPLS Environments	43
	VLANs Within a Bridge Domain or VPLS Instance	43
	Packet Flow Through a Bridged Network with Normalized VLANs	44
	Configuring a Normalized VLAN for Translation or Tagging	45
	Implicit VLAN Translation to a Normalized VLAN	45
	Sending Tagged or Untagged Packets over VPLS Virtual Interfaces	46
	Configuring a Normalized VLAN	46
	Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces	47
	Example: Configuring a Provider Bridge Network with Normalized VLAN Tags	47
	Example: Configuring a Provider VPLS Network with Normalized VLAN Tags	51
	Example: Configuring One VPLS Instance for Several VLANs	55
Chapter 5	Bulk Administration of Layer 2 Features on MX Series Routers	63
	Bulk Configuration of VLANs and Bridge Domains	63
	Example: Configuring VLAN Translation with a VLAN ID List	63
	Example: Configuring Multiple Bridge Domains with a VLAN ID List	64
Chapter 6	Dynamic Profiles for VLAN Interfaces and Protocols	67
	Dynamic Profiles for VPLS Pseudowires	67
	Example: Configuring VPLS Pseudowires with Dynamic Profiles—Basic Solutions	68
	VPLS Pseudowire Interfaces Without Dynamic Profiles	68
	VPLS Pseudowire Interfaces and Dynamic Profiles	69
	CE Routers Without Dynamic Profiles	70
	CE Routers and Dynamic Profiles	71
	Example: Configuring VPLS Pseudowires with Dynamic Profiles—Complex Solutions	72
	Configuration of Routing Instance and Interfaces Without Dynamic Profiles	73
	Configuration of Routing Instance and Interfaces Using Dynamic Profiles	74
	Configuration of Tag Translation Using Dynamic Profiles	76
Chapter 7	MX Series Router as a DHCP Relay Agent	79
	MX Series Router as a Layer 2 DHCP Relay Agent	79
	Example: Configuring DHCP Relay in a Bridge Domain VLAN Environment	80
	Example: Configuring DHCP Relay in a VPLS Routing Instance Environment	81
Chapter 8	MX Series Router in an ATM Ethernet Interworking Function	83
	MX Series Router ATM Ethernet Interworking Function	83
	Example: Configuring MX Series Router ATM Ethernet Interworking	85
	Configuring Router PE2 with a Layer 2 Circuit	86
	Configuring Router PE2 with a Layer 2 Circuit over Aggregated Ethernet	88
	Configuring Router PE2 with a Remote Interface Switch	91
	Configuring Router PE2 with a Remote Interface Switch over Aggregated Ethernet	94

Part 3	Ethernet Filtering, Monitoring, and Fault Management Solutions for MX Series Routers	
Chapter 9	Layer 2 Firewall Filters	101
	Firewall Filters for Bridge Domains and VPLS Instances	101
	Example: Configuring Policing and Marking of Traffic Entering a VPLS Core	102
	Example: Configuring Filtering of Frames by MAC Address	104
	Example: Configuring Filtering of Frames by IEEE 802.1p Bits	106
	Example: Configuring Filtering of Frames by Packet Loss Priority	107
Chapter 10	IEEE 802.1ag OAM Connectivity-Fault Management	109
	Ethernet Operations, Administration, and Maintenance	109
	Ethernet OAM Connectivity Fault Management	110
	Example: Configuring Ethernet CFM over VPLS	111
	Example: Configuring Ethernet CFM on Bridge Connections	118
	Example: Configuring Ethernet CFM on Physical Interfaces	122
Chapter 11	ITU-T Y.1731 Ethernet Frame Delay Measurements	125
	Ethernet Frame Delay Measurements	125
	Configuring MEP Interfaces to Support Ethernet Frame Delay Measurements	128
	Triggering an Ethernet Frame Delay Measurements Session	129
	Viewing Ethernet Frame Delay Measurements Statistics	130
	Example: Configuring One-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces	131
	Example: Configuring Two-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces	136
	Example: Configuring Ethernet Frame Delay Measurements with Untagged Interfaces	140
Chapter 12	IEEE 802.3ah OAM Link-Fault Management	143
	Ethernet OAM Link Fault Management	143
	Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge	144
	Example: Configuring Ethernet LFM for CCC	145
	Example: Configuring Ethernet LFM for Aggregated Ethernet	146
	Example: Configuring Ethernet LFM with Loopback Support	148
Chapter 13	Ethernet Ring Protection	151
	Ethernet Ring Protection	151
	Ethernet Ring Protection Using Ring Instances for Load Balancing	153
	Example: Configuring Ethernet Ring Protection for MX Series Routers	154
	Example Topology	154
	Router 1 (RPL Owner) Configuration	155
	Router 2 Configuration	157
	Router 3 Configuration	158
	Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers	160
	Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation	177
	Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition	179

Part 4

Index

Index	185
-------------	-----

List of Figures

Part 1	Overview	
Chapter 1	Overview of Ethernet Solutions	3
	Figure 1: Native (Normal) and VLAN-Tagged Ethernet Frames	12
	Figure 2: A Metro Ethernet Network	15
	Figure 3: A Metro Ethernet Network with MX Series Routers	16
	Figure 4: VLAN Tags on a Metro Ethernet Network	16
Part 2	Basic Solutions for MX Series Routers	
Chapter 2	Basic Layer 2 Features on MX Series Routers	21
	Figure 5: Bridging Network with MX Series Routers	23
	Figure 6: Designated, Root, and Alternate Ports	34
Chapter 4	VLANs Within Bridge Domain and VPLS Environments	43
	Figure 7: Provider Bridge Network Using Normalized VLAN Tags	48
	Figure 8: VLAN Tags and VPLS Labels	52
	Figure 9: Many VLANs on One VPLS Instance	56
Chapter 8	MX Series Router in an ATM Ethernet Interworking Function	83
	Figure 10: ATM Ethernet VLAN Interworking	83
	Figure 11: ATM Ethernet VLAN Interworking Packet Structure	84
	Figure 12: CCC-to-Stacked VLAN Translation	84
	Figure 13: ATM Ethernet VLAN Interworking	85
Part 3	Ethernet Filtering, Monitoring, and Fault Management Solutions for MX Series Routers	
Chapter 9	Layer 2 Firewall Filters	101
	Figure 14: Policing and Marking Traffic Entering a VPLS Core	102
Chapter 10	IEEE 802.1ag OAM Connectivity-Fault Management	109
	Figure 15: Ethernet OAM with VPLS	112
	Figure 16: Ethernet CFM over a Bridge Network	118
	Figure 17: Ethernet CFM on Physical Interfaces	122
Chapter 11	ITU-T Y.1731 Ethernet Frame Delay Measurements	125
	Figure 18: Ethernet OAM Overview	126
Chapter 12	IEEE 802.3ah OAM Link-Fault Management	143
	Figure 19: Ethernet LFM Between Provider Edge and Customer Edge	144
	Figure 20: Ethernet LFM for CCC	145
	Figure 21: Ethernet LFM for Aggregated Ethernet	147

Chapter 13

Figure 22: Ethernet LFM with Loopback Support	148
Ethernet Ring Protection	151
Figure 23: Ethernet Ring Protection Example Nodes	154
Figure 24: ERP with Multiple Protection Instances Configured on Three MX Series Routers	162

List of Tables

	About This Guide	xiii
	Table 1: Notice Icons	xvii
	Table 2: Text and Syntax Conventions	xvii
Part 3	Ethernet Filtering, Monitoring, and Fault Management Solutions for MX Series Routers	
Chapter 11	ITU-T Y.1731 Ethernet Frame Delay Measurements	125
	Table 3: Monitor Ethernet Delay Command Parameters	129
	Table 4: Show Ethernet Delay Command Parameters	131
Chapter 13	Ethernet Ring Protection	151
	Table 5: Components of the Network Topology	162

About This Guide

This preface provides the following guidelines for using the *Junos[®] OS MX Series 3D Universal Edge Routers Solutions Guide*:

- [Junos Documentation and Release Notes on page xiii](#)
- [Objectives on page xiv](#)
- [Audience on page xiv](#)
- [Supported Routing Platforms on page xv](#)
- [Using the Indexes on page xv](#)
- [Using the Examples in This Manual on page xv](#)
- [Documentation Conventions on page xvi](#)
- [Documentation Feedback on page xviii](#)
- [Requesting Technical Support on page xviii](#)

Junos Documentation and Release Notes

For a list of related Junos documentation, see <http://www.juniper.net/techpubs/software/junos/>.

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

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

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at <http://www.juniper.net/books>.

Objectives

This guide provides an overview of the Layer 2 features of the Junos OS and describes how to configure the features to provide solutions to several network scenarios.



NOTE: For additional information about the Junos OS—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 Layer 2 features of the Junos OS.

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 OS currently supports the following routing platforms:

- Juniper Networks MX Series 3D Universal Edge Routers

Using the Indexes

This reference contains a standard index with topic entries.

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
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.xml; }
```

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons




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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: <code>user@host> configure</code>
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	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 <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> 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.	<code>stub <default-metric metric>;</code>

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

Convention	Description	Examples
(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 <i>(string1 string2 string3)</i>
# (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]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; 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.	<ul style="list-style-type: none"> 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 .

Documentation Feedback

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

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract,

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Overview of Ethernet Solutions on page 3](#)

CHAPTER 1

Overview of Ethernet Solutions

This chapter discusses the following topics:

- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Ethernet Terms 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 topic provides a list of all the terms and acronyms used in the Junos OS Layer 2 Configuration Guide, 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.
- 802.3ag—The IEEE specification for a wide range of Ethernet Operations, Administration, and Maintenance (OAM) features. See also *OAM*, *CFM*, and *ETH-DM*.
- 802.3ah—The IEEE specification for link fault management (LFM), a method for 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.
- CFM—Connectivity-fault management. The part of Ethernet OAM that monitors events at levels above the physical level, as does LFM. See also *OAM*, *LFM*, and *ETH-DM*.
- 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*.
- ETH-DM—Ethernet Frame Delay Measurements. See also *OAM*, *CFM*, and *Y.1731*.
- 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 bridging and Layer 3 routing within the same bridge domain. Packets arriving on an interface of the bridge domain are Layer 2 switched or Layer 3 routed based on the destination MAC address. Packets addressed to the router's MAC address are routed to other Layer 3 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.
- **Q-in-Q**—See *802.1ad*.
- **PBBN**—Provider backbone bridged network.
- **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**—Virtual LAN. 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*.
- **Y.1731**—The international standard for Ethernet Frame Delay Measurements (ETH-DM).

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.

Related Documentation

- [Ethernet Networking](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)

- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

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 (Layer 1), the data link or MAC layer (Layer 2), and the network layer (Layer 3).



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), Layer 2 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 (Layer 3) 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 Layer 2. 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.

Related Documentation

- Ethernet Networking
- [Ethernet Terms and Acronyms on page 3](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Network Addressing at Layer 2 and Layer 3

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 Layer 2 frames and Layer 3 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 a public IP address 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.

- Related Documentation**
- [Ethernet Networking](#)
 - [Ethernet Terms and Acronyms on page 3](#)
 - [Networking and Internetworking with Bridges and Routers on page 6](#)
 - [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
 - [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
 - [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
 - [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
 - [Networking at Layer 2: Logical Interface Types on page 14](#)
 - [A Metro Ethernet Network with MX Series Routers on page 15](#)
 - [Layer 2 Networking Standards on page 17](#)

Networking at Layer 2: Benefits of Ethernet Frames

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.

.....

**Related
Documentation**

- Ethernet Networking
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Networking at Layer 2: Challenges of Ethernet MAC Addresses

If a networked Layer 2 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 Layer 2 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 Layer 2 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).

**Related
Documentation**

- Ethernet Networking
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)

- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Networking at Layer 2: Forwarding VLAN Tagged Frames

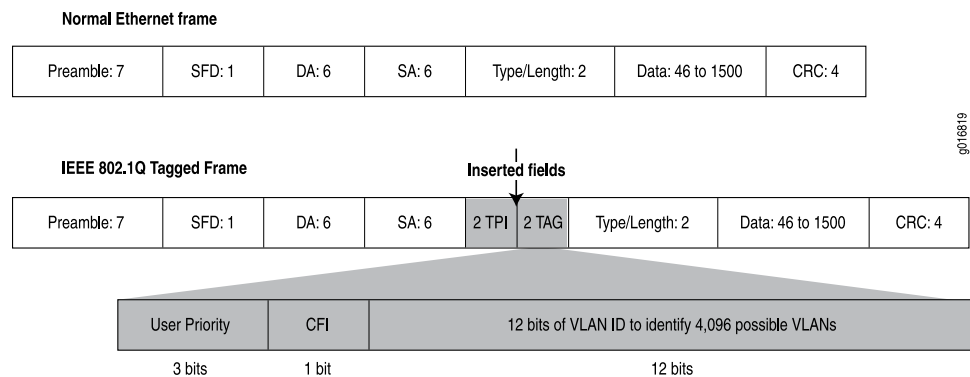
VLAN tags were not developed as a way to limit network node table entries. They were originally invented to allow LAN switches to distinguish between physical groups of LAN ports and logical groups of LAN ports. In other words, there was a need to configure a LAN switch (or group of local LAN switches) 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 12](#), 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 Frames


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.

Related Documentation

- [Ethernet Networking](#)
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Networking at Layer 2: Forwarding Dual-Tagged Frames

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.

- Related Documentation**
- Ethernet Networking
 - [Ethernet Terms and Acronyms on page 3](#)

- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

Networking at Layer 2: Logical Interface Types

Two main types of interfaces are used in Layer 2 configurations:

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



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

Related Documentation

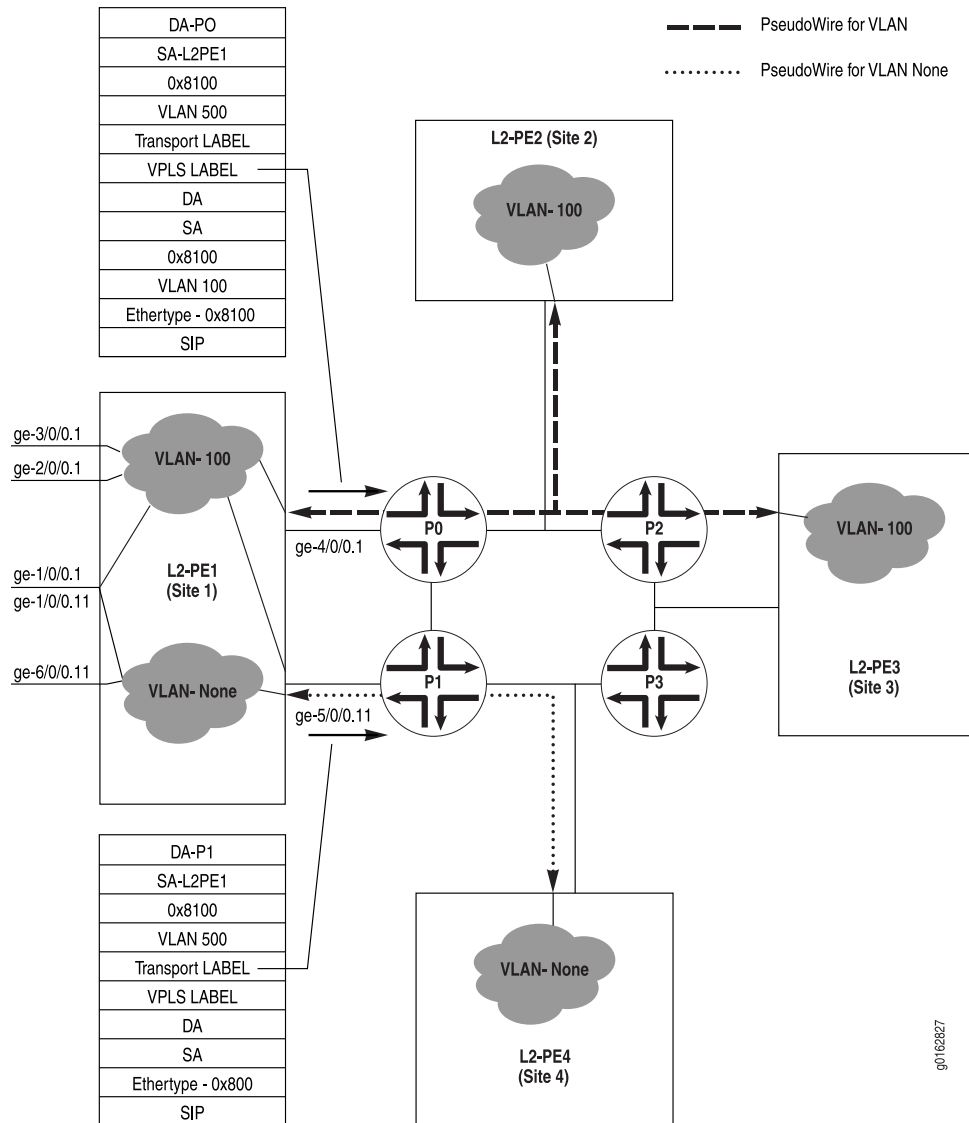
- [Ethernet Networking](#)
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)
- [Layer 2 Networking Standards on page 17](#)

A Metro Ethernet Network with MX Series Routers

What would a Metro Ethernet network with Juniper Networks MX Series 3D Universal Edge Router 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.

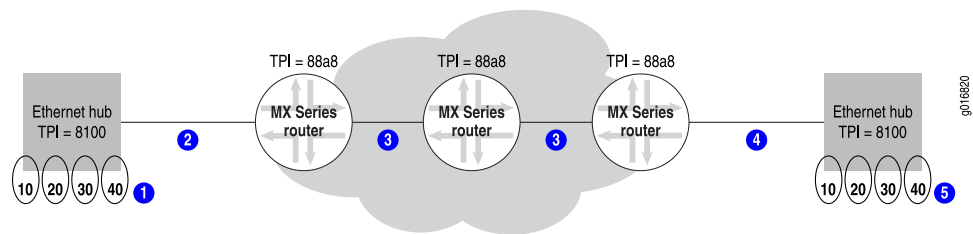
The VLAN labels in the packet are stacked with MPLS labels, as shown in [Figure 2 on page 15](#). For a more detailed examination of this type of Metro Ethernet network, see “[Example: Configuring a Provider VPLS Network with Normalized VLAN Tags](#)” on [page 51](#).

Figure 2: A Metro Ethernet Network



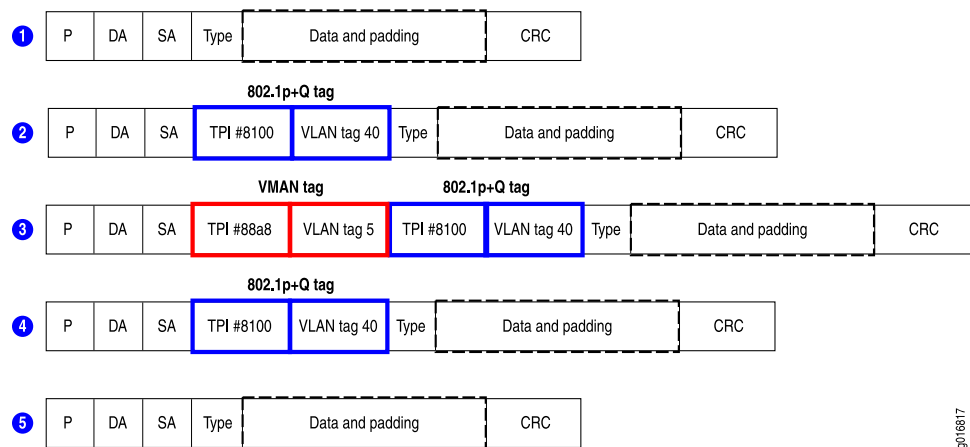
Another possible configuration, this one without the VPLS and MPLS core, is shown in [Figure 3 on page 16](#).

Figure 3: A Metro Ethernet Network with MX Series Routers



In [Figure 3 on page 16](#), 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 16](#).

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 16](#) 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.

Related Documentation

- [Ethernet Networking](#)
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)

- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)
- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [Layer 2 Networking Standards on page 17](#)

Layer 2 Networking Standards

For additional information about the Layer 2 networking features available on Juniper Networks MX Series 3D Universal Edge Router, see the following references:

- 802.1ad—IEEE standard *Provider Bridges*.
- 802.1ag—IEEE standard *Connectivity Fault Management*.
- 802.1ah—IEEE standard *Provider Backbone Bridges*.
- 802.1p—IEEE draft standard *Wireless Access in Vehicular Environments*.
- 802.1Q—IEEE standard *Provider Backbone Bridge Traffic Engineering*.
- 802.3ah-2004—IEEE standard *Operations Administration, and Management (OAM)* for link fault management (LFM), or simple connectivity fault management (CFM) at the data link layer. Also known as “Ethernet in the First Mile (EFM)” and EFM-OAM.
- 802.3-2008, Clause 57—IEEE standard *Operations Administration, and Maintenance (OAM)*. Incorporates 802.3ah-2004 within the IEEE standard *Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*.
- RFC 4761—IETF draft *Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling*.
- RFC 4762—IETF draft *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*.
- Y.1731—ITU-T recommendation *OAM Functions and Mechanisms for Ethernet-based Networks*.
- OSI-RM—*Open Systems Interconnection Reference Model*.

Related Documentation

- [Ethernet Networking](#)
- [Ethernet Terms and Acronyms on page 3](#)
- [Networking and Internetworking with Bridges and Routers on page 6](#)
- [Network Addressing at Layer 2 and Layer 3 on page 7](#)
- [Networking at Layer 2: Benefits of Ethernet Frames on page 9](#)
- [Networking at Layer 2: Challenges of Ethernet MAC Addresses on page 10](#)

- [Networking at Layer 2: Forwarding VLAN Tagged Frames on page 11](#)
- [Networking at Layer 2: Forwarding Dual-Tagged Frames on page 13](#)
- [Networking at Layer 2: Logical Interface Types on page 14](#)
- [A Metro Ethernet Network with MX Series Routers on page 15](#)

PART 2

Basic Solutions for MX Series Routers

This chapter discusses the following topics:

- [Basic Layer 2 Features on MX Series Routers on page 21](#)
- [Virtual Switches on page 39](#)
- [VLANs Within Bridge Domain and VPLS Environments on page 43](#)
- [Bulk Administration of Layer 2 Features on MX Series Routers on page 63](#)
- [Dynamic Profiles for VLAN Interfaces and Protocols on page 67](#)
- [MX Series Router as a DHCP Relay Agent on page 79](#)
- [MX Series Router in an ATM Ethernet Interworking Function on page 83](#)

CHAPTER 2

Basic Layer 2 Features on MX Series Routers

- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)
- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Layer 2 Features for a Bridging Environment

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

Configuring Layer 2 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, Layer 2 filtering). This chapter offers a fairly complex configuration for Layer 2 processing in a bridged environment.

Generally, there are four things that you must configure in an Layer 2 environment:

- Interfaces and virtual LAN (VLAN) tags—Layer 2 interfaces are usually various type of Ethernet links with VLAN tags used to connect to customer devices or other bridges or routers.
- Bridge domains—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.

- 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. (Layer 2 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.

Related Documentation

- Ethernet Networking
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)
- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Example Roadmap: Configuring a Basic Bridge Domain Environment

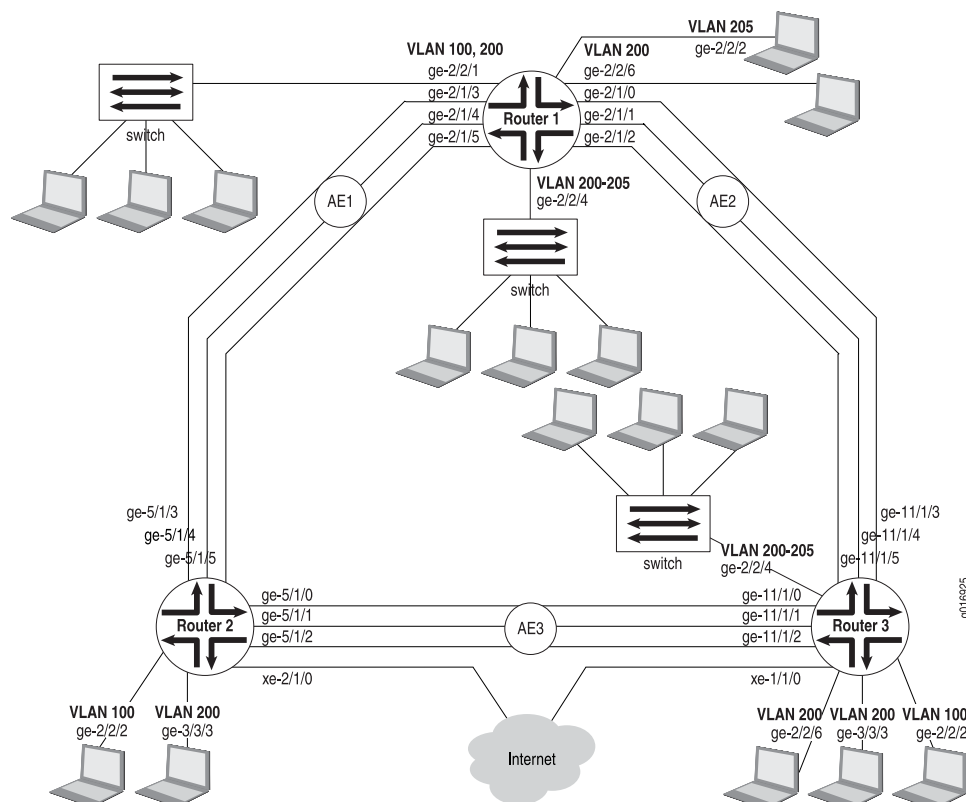
Configuring Layer 2 features on MX Series routers can vary from the very simple (aggregated Ethernet trunk interfaces, spanning trees), to the more complex (inner and outer VLAN tags, broadcast domains), to the very complicated (integrated bridging and routing, Layer 2 filtering). This example offers a fairly complex configuration for Layer 2 processing in a bridged environment.

- [Example Topology on page 22](#)
- [Example Scenario on page 23](#)
- [Example Configuration Summary on page 24](#)

Example Topology

Consider the network in [Figure 5 on page 23](#). The figure shows three MX Series routers acting as Layer 2 devices.

Figure 5: Bridging Network with MX Series Routers



The three routers each have a series of hosts on their Ethernet interfaces, as well as aggregated Ethernet links between them. Router 2 and Router 3 are linked to the Internet, and Router 1 and Router 3 are also linked to switches configured with a range of VLANs, as shown in the figure. Because the VLAN tags are important, 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).

Example Scenario

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`).
- Router 2 and Router 3 have IRB configured so that they can pass traffic to other routers in the rest of the network.

- Router 1 has an access interface which provides bridging on VLAN 205 and is connected to a customer device configured on **ge-2/2/2**. Router 3 has an access interface which provides bridging on VLAN 200 and is connected to a customer device configured on **ge-2/2/6**.
- Router 1 and Router 3 are configured with a trunk interface to a switch for VLANs 200–205. On both routers, this interface is **ge-2/2/4**.

Example Configuration Summary

This procedure summarizes the minimum configuration steps required for Layer 2 processing in a bridged environment, as described in [“Layer 2 Features for a Bridging Environment” on page 21](#). The individual configuration steps are described in greater detail in separate topics.

To configure Layer 2 processing in a bridged domain network:

1. Configure the Ethernet interfaces and VLAN tags on all three routers, as described in [“Example Step: Configuring Interfaces and VLAN Tags” on page 24](#)
2. Configure the bridge domains on all three routers, as described in [“Example Step: Configuring Bridge Domains” on page 30](#).
3. Configure the Spanning Tree Protocol on all three routers, as described in [“Example Step: Configuring Spanning Tree Protocols” on page 32](#)
4. Configure IRB, as described in [“Example Step: Configuring Integrated Bridging and Routing” on page 34](#)

Related Documentation

- Ethernet Networking
- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Example Step: Configuring 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.

To configure the Ethernet interfaces and VLAN tags on all three routers:

1. Configure the Ethernet interfaces and VLAN tags on Router 1:

```
[edit]
chassis {
  aggregated-devices {
    ethernet {
      device-count 2; # Number of AE interfaces on router
    }
  }
}
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-2/2/2 {
  unit 0 {
```

```
        family bridge {
            interface-mode access;
            vlan-id 205;
        }
    }
}
interfaces ge-2/2/4 {
    native-vlan-id 200; # Untagged packets get vlan 200 tag
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 200-205; # This trunk port is part of VLAN range 200–205
        }
    }
}
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 {
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 100, 200–205;
        }
    }
}
```

2. Configure the Ethernet interfaces and VLAN tags on Router 2:

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

3. Configure the Ethernet interfaces and VLAN tags on Router 3:

```
[edit]
chassis {
    aggregated-devices {
        ethernet {
            device-count 2; # Number of AE interfaces on router
        }
    }
}
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-2/2/4 {
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 200-205; # This trunk port is part of VLAN range 200–205
        }
    }
}
interfaces ge-2/2/6 {
    unit 0 {
        family bridge {
            interface-mode access;
            vlan-id 200;
        }
    }
}
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-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 {
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 100, 200–205;
        }
    }
}
interfaces ae3 {
    encapsulation extended-vlan-bridge;
    vlan-tagging;
    unit 100 {
        vlan-id 100;
    }
    unit 200 {
        vlan-id 200;
    }
}
}

```

Related Documentation

- [Ethernet Networking](#)
- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Example Step: Configuring Bridge Domains

There are two supported ways to configure an IRB with bridge domains. One way is to use the **interface-mode trunk** statement. The other way is shown here and is known as the encapsulation method. The encapsulation method is the preferred way to configure an IRB with bridge domains.

To configure the bridge domains on all three routers:

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;
  }
  vlan200 {
    domain-type bridge;
    vlan-id 200;
    interface ge-2/2/1.200;
    interface ge-2/2/6.200;
    interface ae1.200;
  }
  vlan201 {
    domain-type bridge;
    vlan-id 201;
  }
  vlan202 {
    domain-type bridge;
    vlan-id 202;
  }
  vlan203 {
    domain-type bridge;
    vlan-id 203;
  }
  vlan204 {
    domain-type bridge;
    vlan-id 204;
  }
  vlan205 {
    domain-type bridge;
    vlan-id 205;
  }
}
```

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;
    }
}

```

3. Configure a bridge domain on Router 3:

```

[edit]
bridge-domains {
    vlan100 {
        domain-type bridge;
        vlan-id 100;
        interface ge-2/2/2.100;
        interface ae3.100;
    }
    vlan200 {
        domain-type bridge;
        vlan-id 200;
        interface ge-3/3/3.200;
        interface ae3.200;
    }
    vlan201 {
        domain-type bridge;
        vlan-id 201;
    }
    vlan202 {
        domain-type bridge;
        vlan-id 202;
    }
    vlan203 {
        domain-type bridge;
        vlan-id 203;
    }
    vlan204 {
        domain-type bridge;
        vlan-id 204;
    }
    vlan205 {
        domain-type bridge;
        vlan-id 205;
    }
}

```

Related Documentation

- Ethernet Networking
- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)

- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Example Step: 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).

To configure the Spanning Tree Protocol on all three routers:

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
    }
  }
}
```

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
    }
  }
}
```

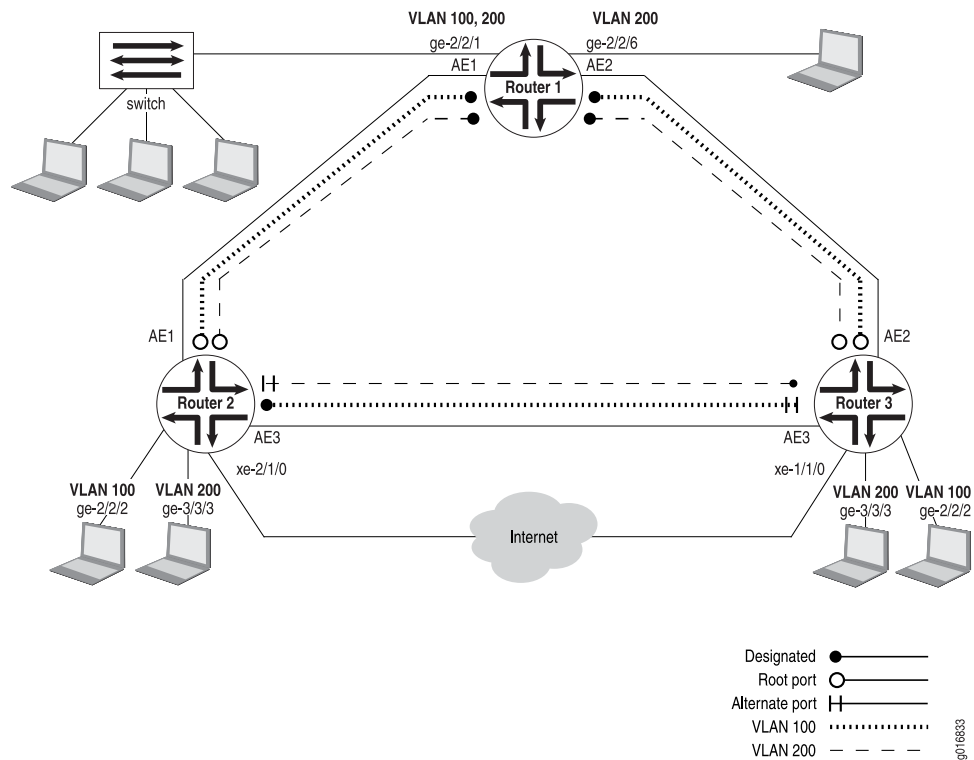
```
msti 2 {  
    vlan200; # This VLAN corresponds to MSTP instance 2  
}  
}  
}
```

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 34](#).

Figure 6: Designated, Root, and Alternate Ports



Related Documentation

- [Ethernet Networking](#)
- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)
- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Integrated Bridging and Routing on page 34](#)

Example Step: Configuring Integrated Bridging and Routing

Router 2 and Router 3 on the bridging network act as a kind of gateway to the Layer 3 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.



NOTE: You 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.

To configure IRB on Router 2 and Router 3:



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 ae1 and ae3 interfaces have been previously configured in [“Example Step: Configuring Interfaces and VLAN Tags”](#) on page 24.

1. Configure the router link and IRB on Router 2:

```
[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;
        }
      }
    }
  }
}
```

```
    }  
  }  
  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;  
    }  
  }  
}
```

2. Configure the router link and IRB on Router 3:

```
[edit]  
interfaces {  
  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;  
          }  
        }  
      }  
    }  
    unit 2 {  
      family inet {  
        address 10.0.3.2/24 {
```



```
    }  
  }  
  unit 3 {  
    family inet {  
      address 10.0.3.3/24 {  
      }  
    }  
  }  
  unit 4 {  
    family inet {  
      address 10.0.3.4/24 {  
      }  
    }  
  }  
  unit 5 {  
    family inet {  
      address 10.0.3.5/24 {  
      }  
    }  
  }  
  unit 6 {  
    family inet {  
      address 10.0.3.6/24 {  
      }  
    }  
  }  
  unit 7 {  
    family inet {  
      address 10.0.3.7/24 {  
      }  
    }  
  }  
  unit 8 {  
    family inet {  
      address 10.0.3.8/24 {  
      }  
    }  
  }  
}  
}  
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;  
  }  
  vlan201 {  
    vlan-id 201;  
    routing-interface irb.2  
  }  
  vlan202 {
```

```
        vlan-id 202;  
        routing-interface irb.3  
    }  
    vlan203 {  
        vlan-id 203;  
        routing-interface irb.4  
    }  
    vlan204 {  
        vlan-id 204;  
        routing-interface irb.5  
    }  
    vlan205 {  
        vlan-id 205;  
        routing-interface irb.6  
    }  
}
```

**Related
Documentation**

- Ethernet Networking
- [Layer 2 Features for a Bridging Environment on page 21](#)
- [Example Roadmap: Configuring a Basic Bridge Domain Environment on page 22](#)
- [Example Step: Configuring Interfaces and VLAN Tags on page 24](#)
- [Example Step: Configuring Bridge Domains on page 30](#)
- [Example Step: Configuring Spanning Tree Protocols on page 32](#)

CHAPTER 3

Virtual Switches

This chapter discusses the following topics:

- [Layer 2 Features for a Switching Environment on page 39](#)
- [Configuring Virtual Switches as Separate Routing Instances on page 40](#)

Layer 2 Features for a Switching Environment

Juniper Networks MX Series 3D Universal Edge Routers include all standard Ethernet capabilities as well as enhanced mechanisms for service providers to provision and support large numbers of Ethernet services in addition to all Layer 3 services. The MX Series routers include several features to contain and control the Ethernet environment.

One of these features is the virtual switch. MX Series routers allow the collapsing of multiple diverse switch networks to a single platform by running virtual instances of as many Spanning Tree Protocols (STPs) as needed to support all broadcast domains. This is important because there are many incompatible versions of STP, and without a way to run multiple virtual instances, a separate switch would be needed to support each one. With MX Series virtual switch configuration, you can continue to running existing STP protocols with the option to migrate to a common STP protocol if desired.

Virtual switches also make it easy to separate independent switched Ethernet networks, each possibly carrying several VLANs. Because the same VLAN ID can be used in multiple switched networks, virtual switches can keep each VLAN and broadcast domain logically separated.



NOTE: In a router environment, there is always a default routing instance. When you need only one routing instance on the router, you use the default routing instance without qualification. However, if you need more than one routing instance, you must configure statements to create additional routing instances. In a switching environment, the same is true of virtual switches: if you need more than one virtual switch in addition to the “default,” you must create them.

For more information about STPs and virtual switches, see the Junos OS Layer 2 Configuration Guide.

- Related Documentation**
- Ethernet Networking
 - [Configuring Virtual Switches as Separate Routing Instances on page 40](#)

Configuring Virtual Switches as Separate Routing Instances

You can configure two virtual switches as separate routing instances on an MX Series router with bridge domains and VLANs.

Before you begin, you should have already configured a basic bridge domain environment. For a general description of a basic bridge domain environment, see [“Layer 2 Features for a Bridging Environment” on page 21](#). For an example of a basic bridge domain configuration, see [“Example Roadmap: Configuring a Basic Bridge Domain Environment” on page 22](#). More detailed examples are also provided for the four features generally required in a Layer 2 environment:

- Interfaces and VLAN tags required.
- Bridge domains required by the topology.
- Spanning tree protocols required by the topology.
- Integrated bridging and routing required by the topology.

At the end of this configuration, you create two virtual switches as separate routing instances to separate the VLANs and broadcast domains. Because the same VLAN ID can be used in multiple switched networks, virtual switches can keep each VLAN and broadcast domain logically separated.

To configure two virtual switches as separate routing instances:

1. The following statements configure the first virtual switch in a routing instance.

```
[edit]
routing-instances {
  virtual-switch-1 {
    instance-type virtual-switch;
    ...virtual-switch-1 configuration with one STP/VLAN ID set...
  }
}
```

2. The following statement configure the second virtual switch in a different routing instance.

```
[edit]
routing-instances {
  virtual-switch-2 {
    instance-type virtual-switch;
    ...virtual-switch-2 configuration with another STP/VLAN ID set...
  }
}
```

This is not a complete configuration.

For more information about configuring virtual switches, see the Junos OS Layer 2 Configuration Guide.

**Related
Documentation**

- Ethernet Networking
- [Layer 2 Features for a Switching Environment on page 39](#)

CHAPTER 4

VLANs Within Bridge Domain and VPLS Environments

This chapter discusses the following topics:

- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Configuring a Normalized VLAN for Translation or Tagging on page 45](#)
- [Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces on page 47](#)
- [Example: Configuring a Provider Bridge Network with Normalized VLAN Tags on page 47](#)
- [Example: Configuring a Provider VPLS Network with Normalized VLAN Tags on page 51](#)
- [Example: Configuring One VPLS Instance for Several VLANs on page 55](#)

VLANs Within a Bridge Domain or VPLS Instance

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 identifier tags specify the normalized VLAN. For this case, the terms “learn VLAN” and “normalized VLAN” can be used interchangeably.

You can specify the normalized VLAN using one of the following conditions:

- The VLAN identifier is determined explicitly by configuration
- The VLAN identifier is specified as “none,” meaning the VLAN tags are not translated or generated
- The inner and outer VLAN identifier tags are both determined explicitly by configuration

Related Documentation

- [Ethernet Networking](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Configuring a Normalized VLAN for Translation or Tagging on page 45](#)
- [Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces on page 47](#)

- [Example: Configuring a Provider Bridge Network with Normalized VLAN Tags on page 47](#)
- [Example: Configuring a Provider VPLS Network with Normalized VLAN Tags on page 51](#)
- [Example: Configuring One VPLS Instance for Several VLANs on page 55](#)

Packet Flow Through a Bridged Network with Normalized VLANs

Packets received over a Layer 2 logical interface for bridging are processed in a strict sequence of steps.

Packets received over a Layer 2 logical interface for bridging when a normalized VLAN is configured with a single or inner and outer VLAN identifier tags 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*.

Related Documentation

- Ethernet Networking
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Configuring a Normalized VLAN for Translation or Tagging on page 45](#)
- [Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces on page 47](#)
- [Example: Configuring a Provider Bridge Network with Normalized VLAN Tags on page 47](#)
- [Example: Configuring a Provider VPLS Network with Normalized VLAN Tags on page 51](#)
- [Example: Configuring One VPLS Instance for Several VLANs on page 55](#)

Configuring a Normalized VLAN for Translation or Tagging

This topic 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 topic. For more information about configuring Ethernet pseudowires as part of VPLS, see the Junos OS Feature Guides.



NOTE: This topic is not intended as a troubleshooting guide. However, you can use it with a broader troubleshooting strategy to identify Juniper Networks MX Series 3D Universal Edge 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® OS Network Interfaces.
- 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 section discusses the following topics:

- [Implicit VLAN Translation to a Normalized VLAN on page 45](#)
- [Sending Tagged or Untagged Packets over VPLS Virtual Interfaces on page 46](#)
- [Configuring a Normalized VLAN on page 46](#)

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 OS 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 OS 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***. For an example of this configuration, see “[Example: Configuring One VPLS Instance for Several VLANs](#)” on page 55.
- **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.

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 bridging (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. For an example of this configuration, see [“Example: Configuring One VPLS Instance for Several VLANs” on page 55](#).

Related Documentation

- Ethernet Networking
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Example: Configuring a Provider Bridge Network with Normalized VLAN Tags on page 47](#)
- [Example: Configuring a Provider VPLS Network with Normalized VLAN Tags on page 51](#)

Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces

A learning domain is a MAC address database to which 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.

To configure bridging for several VLANs using a minimal amount of configuration and switch resources, use the **vlan-id all** configuration statement to implicitly configure multiple learning domains for a bridge domain or VPLS instance:

- For a logical interface with a single VLAN tag, the statement implicitly creates a learning domain for each normalized VLAN of the interface.
- For a logical interface with dual VLAN tags, the statement implicitly creates a learning domain for each inner VLAN (normalized VLAN).

Related Documentation

- Ethernet Networking
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Example: Configuring One VPLS Instance for Several VLANs on page 55](#)

Example: Configuring a Provider Bridge Network with Normalized VLAN Tags

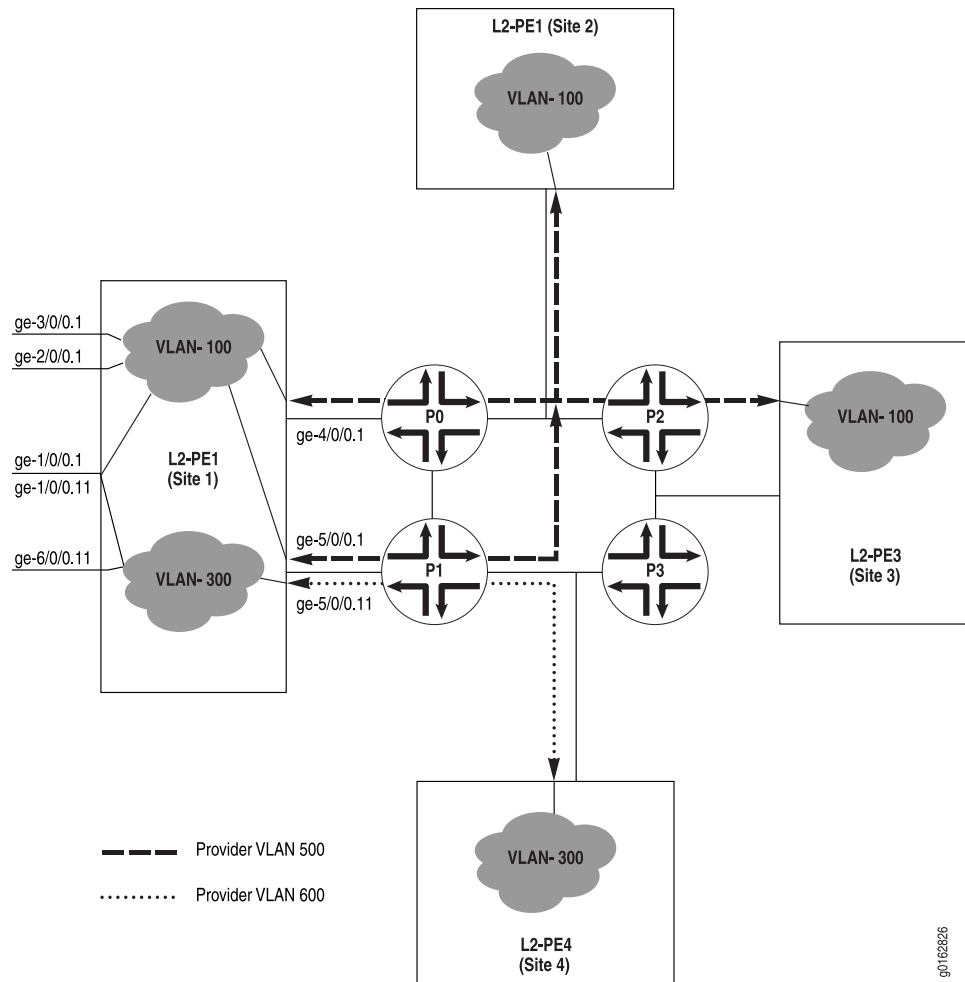
This topic provides a configuration example to help you effectively configure a network of Juniper Networks MX Series 3D Universal Edge Routers for a bridge domain or virtual private LAN service (VPLS) environment. The emphasis here is on choosing the normalized virtual LAN (VLAN) configuration. The VPLS configuration is not covered in this chapter. For more information about configuring Ethernet pseudowires as part of VPLS, see the Junos OS Feature Guides.



NOTE: This topic 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.

Consider the provider bridge network shown in [Figure 7 on page 48](#).

Figure 7: Provider Bridge Network Using Normalized VLAN Tags



The Layer 2 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 value
  }
}
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**.

Related Documentation

- [Ethernet Networking](#)
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Configuring a Normalized VLAN for Translation or Tagging on page 45](#)

Example: Configuring a Provider VPLS Network with Normalized VLAN Tags

This topic provides a configuration example to help you effectively configure a network of Juniper Networks MX Series 3D Universal Edge Routers for a bridge domain or virtual private LAN service (VPLS) environment. The emphasis here is on choosing the normalized virtual LAN (VLAN) configuration. The VPLS configuration is not covered in this chapter.

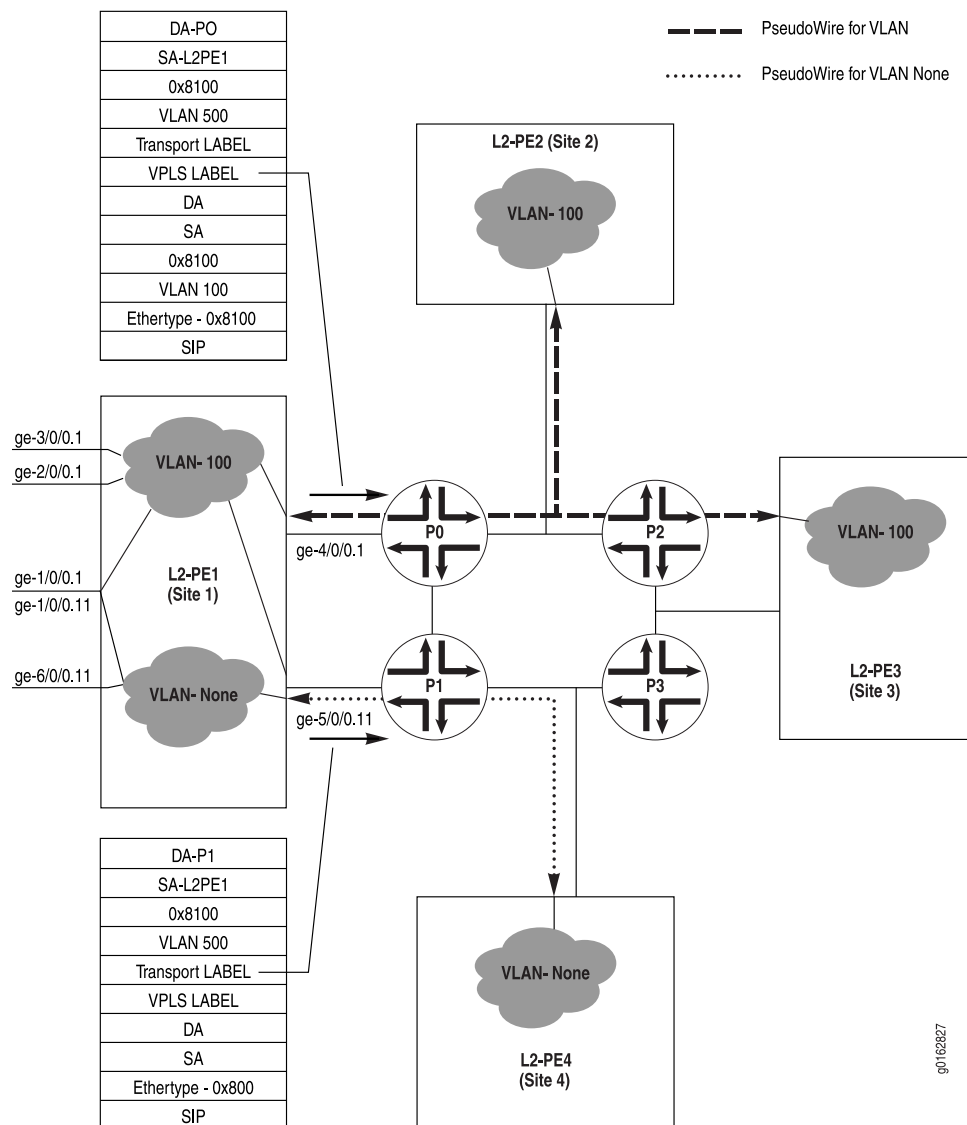
For more information about configuring Ethernet pseudowires as part of VPLS, see the Junos OS Feature Guides.



NOTE: This topic 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.

Consider the VPLS network shown in [Figure 8 on page 52](#).

Figure 8: VLAN Tags and VPLS Labels



The Layer 2 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
```



NOTE: This is not a complete router configuration.

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.

Related Documentation

- Ethernet Networking
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Configuring a Normalized VLAN for Translation or Tagging on page 45](#)

Example: Configuring One VPLS Instance for Several VLANs

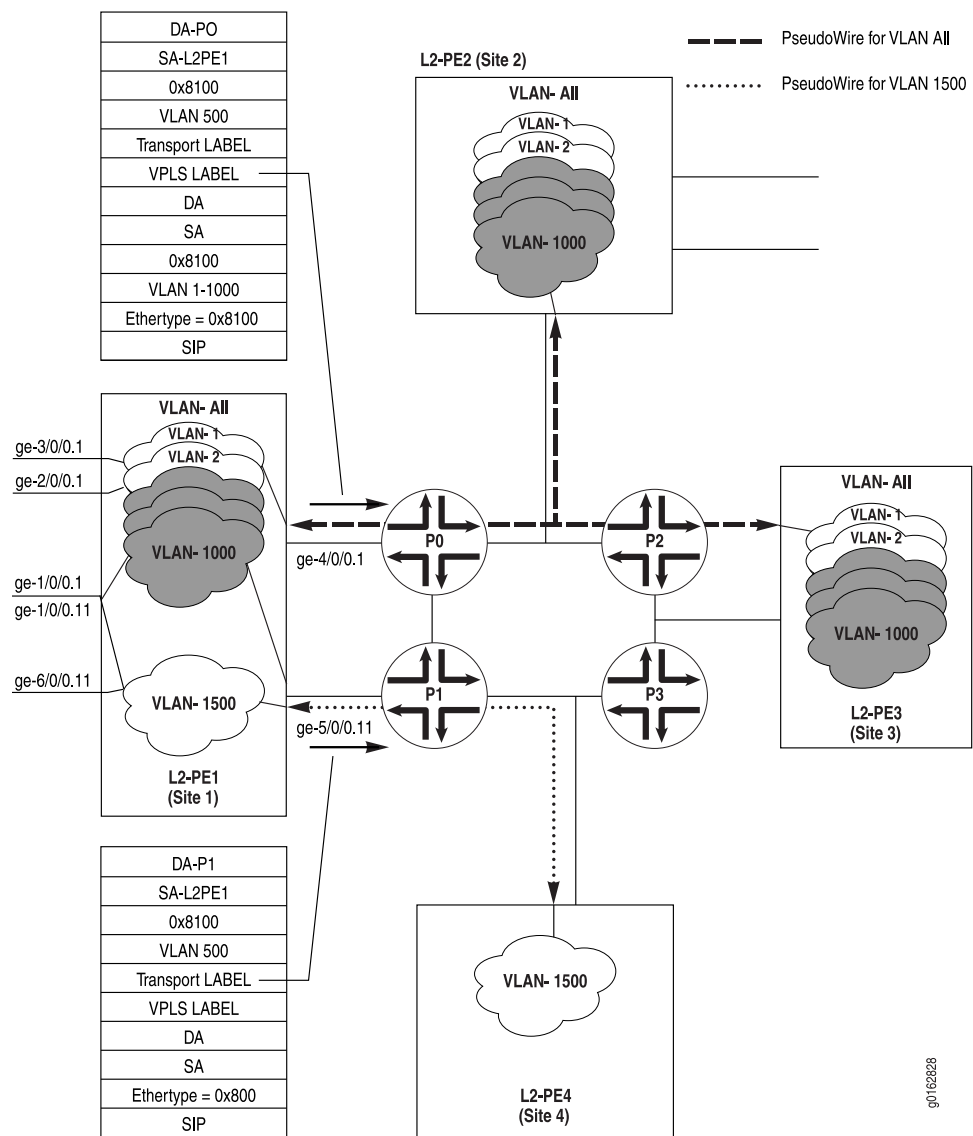
This topic provides a configuration example to help you effectively configure a network of Juniper Networks MX Series 3D Universal Edge Routers for a bridge domain or virtual private LAN service (VPLS) environment. The emphasis here is on choosing the normalized virtual LAN (VLAN) configuration. The VPLS configuration is not covered in this chapter. For more information about configuring Ethernet pseudowires as part of VPLS, see the Junos OS Feature Guides.



NOTE: This topic 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.

Consider the VPLS network shown in [Figure 9 on page 56](#).

Figure 9: Many VLANs on One VPLS Instance



The Layer 2 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.

Alternatively, instead of configuring a VPLS instance, you can define a virtual switch with a bridge domain and associate the logical interfaces as trunk ports with the bridge domain. This configuration is necessary if you want to configure a list or range of VLAN IDs on the logical interfaces and use the **vlan-id all** statement to normalize VLANs.

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

You can configure VPLS ports in a virtual switch so that the logical interfaces of the Layer 2 bridge domains in the virtual switch can handle VPLS routing instance traffic. VPLS configuration no longer requires a dedicated routing instance of type vpls. Packets received on a Layer 2 trunk interface are forwarded within a bridge domain that has the same VLAN identifier. A trunk interface is implicitly associated with bridge domains based on VLAN membership.

You can use either of the following two mechanisms to normalize VLAN identifiers and process them in a bridge domain or a VPLS routing instance:

- By using the **input-vlan-map** and the **output-vlan-map** statements at the **[edit interfaces interface-name]** hierarchy level to configure VLAN mapping.
- By using either the **vlan-id** statement or the **vlan-tags** statement to configure a normalizing VLAN identifier.

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

- Translate, or normalize, the VLAN tags of received packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a learn VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the learn VLAN identifier.

If you configure the **vlan-id all** statement in a VPLS routing instance, we recommend using the **input-vlan-map pop** and **output-vlan-map push** statements on the logical interface to pop the service VLAN ID on input and push the service VLAN ID on output and in this way limit the impact of doubly-tagged frames on scaling. You cannot use the native **vlan-id** statement when the **vlan-id all** statement is included in the configuration.

For the same network topology illustrated in [Figure 9 on page 56](#), if VLANs 1 through 1000 for customer C1 span the same sites, you can normalize the VLANs by doing one of the following. Using either of these optimal methods, you can switch and normalize all of these VLANs in an effective, streamlined manner without configuring separately for each VLAN ID.

- By configuring a VPLS routing instance if the logical interfaces are specified with a range of consecutive VLANs or a list of non-contiguous VLAN IDs and using VLAN maps to rewrite the VLAN tags on all of the incoming and outgoing packets on the logical interfaces with a normalized VLAN ID
- By configuring a virtual-switch instance consisting of a set of bridge domains that are associated with one or more logical interfaces configured as a trunk port

You cannot use the **vlan-id** statement to enable VLAN normalization in VPLS instances, if the logical interfaces in the VPLS instance are configured with the **vlan-id-list** or **vlan-id-range** statement. In such a scenario, you can use the **input-vlan-map** or the **output-vlan-map** option to achieve VLAN normalization.

The following example illustrates the use of the VLAN mapping functionality in VPLS routing instances to normalize VLANs. This method is beneficial in scenarios with flexible VLAN tagging (asymmetric tag depth). In such an environment, the VLAN configuration data that you specified applies the appropriate VLAN tags to the input and output VLAN maps for the ingress and egress logical interfaces respectively. For example, if certain packets are received as single- tagged packets and if the remaining packets are received as double-tagged packets, using VLAN mapping enables normalization.

Using the VLAN mapping capability is effective only if packets of unequal VLAN tags are received or transmitted from logical interfaces to achieve normalization. We recommend that you do not use VLAN mapping in environments in which the VLAN tags are of equal tag depths for optimal configuration. In such cases, you can use the **vlan-id all** statement to enable normalization of VLANs.

```
[edit]
interfaces ge-1/0/0 {
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 1 {
    encapsulation vlan-vpls;
    vlan-id-range 1-1000;
    input-vlan-map {
      push; /* Push the service vlan on input */
      vlan-id 1200; # This VLAN ID is the normalized VLAN for incoming packets
    }
    output-vlan-map pop; /* Pop the service vlan on output */
  }
  unit 11 {
    encapsulation vlan-vpls;
    vlan-id 1500;
  }
}
interfaces ge-2/0/0 {
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 1 {
```

```

encapsulation vlan-vpls;
vlan-id-range 1-1000; # Note the use of the VLAN id range statement.
input-vlan-map {
    push; /* Push the service vlan on input */
    vlan-id 1300; # This VLAN ID is the normalized VLAN for incoming packets
}
output-vlan-map pop; /* Pop the service vlan on output */
}
}
}
interfaces ge-3/0/0 {
    encapsulation flexible-ethernet-services;
    flexible-vlan-tagging;
    unit 1 {
        encapsulation vlan-vpls;
        vlan-id 1-1000;
        input-vlan-map {
            push; /* Push the service vlan on input */
            vlan-id 1400; # This VLAN ID is the normalized VLAN for incoming packets
        }
        output-vlan-map pop; /* Pop the service vlan on output */
    }
}
}
interfaces ge-6/0/0 {
    encapsulation flexible-ethernet-services;
    flexible-vlan-tagging;
    unit 11 {
        encapsulation vlan-vpls;
        vlan-id 1500;
    }
}
routing-instances {
    customer-c1-v1-to-v1000 {
        instance-type vpls;
        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;
        interface ge-1/0/0.11;
        interface ge-6/0/0.11;
    } # End of customer-c1-v1500
} # End of routing-instances

```

The following operations are performed when you use the VLAN mapping 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.
- The VLAN tags of a received packet are compared with the normalized VLAN tags specified with the `vlan-id` statement in the input VLAN map. If the VLAN tags of the received packet are different from the normalized VLAN tags, then the received VLAN tag is converted to the normalized VLAN tag of 1200 for packets received on the logical interface **ge-1/0/0.1**. Similarly, for logical interface **ge-2/0/0.1**, the normalized VLAN

tag is 1300 for received packets and for logical interface **ge-3/0/0.1**, the normalized VLAN tag is 1400. Then, the source MAC address of a received packet is learned based on the normalized VLAN configuration.

For output packets, based on the **output vlan-map pop** statement configured on the logical interfaces **ge-1/0/0.1**, or **ge-2/0/0.1**, or **ge-3/0/0.1**, if the VLAN tags associated with an egress logical interface do not match the normalized VLAN tags within the packet, then the VLAN tags in the packets that are being transmitted from the egress logical interface are removed.

- 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.
- The **input-vlan-map pop** and **output-vlan-map push** statements on the logical interface cause the service VLAN ID to be popped on input and the service VLAN ID to be pushed on output and in this way, the impact of doubly-tagged frames on scaling is limited.

The following example illustrates the use of the **vlan-id all** statement in logical interfaces when a virtual switch instance with a bridge domain is associated with the logical interfaces. You can normalize VLANs and create learning domains for each VLAN. A routing instance uses a trunk bridge to connect different departments in an organization, each with their own VLANs, at two different sites. You must configure a bridge domain and VLAN identifier for each VLAN associated with the trunk interface.

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



```

}
interfaces ge-3/0/0 {
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 1 {
    encapsulation vlan-vpls;
    interface-mode trunk;
    vlan-id-range 1-1000; # Note the use of the VLAN id range statement.
  }
}
}
interfaces ge-6/0/0 {
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 11 {
    encapsulation vlan-vpls;
    interface-mode trunk;
    vlan-id 1500;
  }
}
}
routing-instances {
  customer-c1-virtual-switch {
    instance-type virtual-switch;
    interface ge-1/0/0.1;
    interface ge-2/0/0.1;
    interface ge-3/0/0.1;
    bridge-domains {
      c1-vlan-v1-to-v1000 {
        vlan-id all; # Note the use of the VLAN id all statement
      }
    }
  } # End of customer-c1-v1-to-v1000
  customer-c2-virtual-switch {
    instance-type virtual-switch;
    interface ge-1/0/0.11;
    interface ge-6/0/0.11;
    bridge-domains {
      c1-vlan-v1500 {
        vlan-id all; # Note the use of the VLAN id all statement
      }
    }
  } # End of customer-c1-v1500
} # End of routing-instances

```

Note the use of the **vlan-id all** and **vlan-id-range** statements in the virtual-switch 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 operations are performed when you use the **vlan-id all** configuration:

- The logical interfaces are configured as a trunk port that multiplexes traffic from multiple VLANs and usually interconnects switches.
- All the VLAN identifiers are associated with a Layer 2 trunk port. Each of the logical interfaces, **ge-1/0/0.1**, or **ge-2/0/0.1**, or **ge-3/0/0.1**, accepts packets tagged with any VLAN ID specified in the respective **vlan-id-range** statements.
- 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 all** configuration within the bridge domain **c1-vlan-v1-to-v1000** for **customer-c1-virtual-switch** sets the normalized VLAN value. For a logical interface with a single VLAN tag, a learning domain implicitly created for each normalized VLAN of the interface.
- Bridge domain **c1-vlan-v1-to-v1000** for **customer-c1-virtual-switch** has three 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**.
- Packets received on logical interfaces **ge-1/0/0.1** or **ge-6/0/0.1** with a single VLAN tag of 1500 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-virtual-switch**, 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.

**Related
Documentation**

- Ethernet Networking
- [VLANs Within a Bridge Domain or VPLS Instance on page 43](#)
- [Packet Flow Through a Bridged Network with Normalized VLANs on page 44](#)
- [Configuring Learning Domains for VLAN IDs Bound to Logical Interfaces on page 47](#)

CHAPTER 5

Bulk Administration of Layer 2 Features on MX Series Routers

This chapter discusses the following topics:

- [Bulk Configuration of VLANs and Bridge Domains on page 63](#)
- [Example: Configuring VLAN Translation with a VLAN ID List on page 63](#)
- [Example: Configuring Multiple Bridge Domains with a VLAN ID List on page 64](#)

Bulk Configuration of VLANs and Bridge Domains

In some cases, service providers must deal with thousands of bridge domains on a single switch. By default the router does not create more than one bridge domain. The configuration of even several hundred bridge domains one at a time can be a burden.

However, you can configure multiple bridge domains with only one statement. Each bridge domain will have the form **prefix-vlan-number**. The prefix and number are supplied by the configuration statement.

Related Documentation

- [Ethernet Networking](#)
- [Example: Configuring VLAN Translation with a VLAN ID List on page 63](#)
- [Example: Configuring Multiple Bridge Domains with a VLAN ID List on page 64](#)

Example: Configuring VLAN Translation with a VLAN ID List

In many cases, the VLAN identifiers on the frames of an interface's packets are not exactly correct. VLAN translation, or VLAN rewrite, allows you to configure bidirectional VLAN identifier translation with a list on frames arriving on and leaving from a logical interface. This lets you use unique VLAN identifiers internally and maintain legacy VLAN identifiers on logical interfaces.

To perform VLAN translation on the packets on a trunk interface, insert the **vlan-rewrite** statement at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level. You must also include the **family bridge** statement at the same level because VLAN translation is only supported on trunk interfaces. The reverse translation takes place on egress. In

other words, if VLAN 200 is translated to 500 on ingress, VLAN 500 is translated to VLAN 200 on egress.

The following example translates incoming trunk packets from VLAN identifier 200 to 500 and 201 to 501 (other valid VLAN identifiers are not affected):

```
[edit interfaces ge-1/0/1]
unit 0 {
  ... # Other logical interface statements
  family bridge {
    interface-mode trunk # Translation is only for trunks
    vlan-id-list [ 100 500-600 ];
    vlan-rewrite {
      translate 200 500;
      translate 201 501;
    }
    ... # Other bridge statements
  }
}
```



NOTE: This example also translates frame VLANs from 500 to 200 and 501 to 201 on egress.

**Related
Documentation**

- Ethernet Networking
- [Bulk Configuration of VLANs and Bridge Domains on page 63](#)
- [Example: Configuring Multiple Bridge Domains with a VLAN ID List on page 64](#)

Example: Configuring Multiple Bridge Domains with a VLAN ID List

To configure multiple bridge domains with one statement, include the **vlan-id-list** statement at the **[edit bridge-domains]** hierarchy level.

The following example automatically configures 4093 bridge domains named **sales-vlan-2** through **sales-vlan-4094**:

```
[edit]
bridge-domains {
  sales { # This is the prefix
    vlan-id-list [ 2-4096 ]; # These are the numbers
  }
}
```

You can configure these bridge domains in a virtual switch routing instance. However, if a VLAN identifier is already part of a VLAN identifier list in a bridge domain under a routing instance, then you cannot configure an explicit bridge domain with that VLAN identifier. In other words, there can be no overlap between a VLAN identifier list and another VLAN identifier configuration.

The following example removes the VLAN identifier 5 from the original VLAN list (**vlan-id-list [1–10]**) and configures the bridge domain explicitly:

```
[edit routing-instance rtg-inst-10]
instance-type virtual-switch;
interface ge-7/3/0.0;
bridge-domains {
  bd-vlan-5 {
    vlan-id 5;
  }
  bd {
    vlan-id [ 1–4 6–10 ];
  }
}
```

If a VLAN identifier is already part of a VLAN identifier list in a bridge domain under a routing instance, then you must delete the VLAN identifier from the list before you can configure an explicit or “regular” bridge domain. Also, the explicit bridge domain will not perform properly unless it has the same name as the bridge domain in the VLAN identifier list.

In other words, if **sales-vlan-100** was part of a bridge domain VLAN list and you wish to configure it explicitly, you must use the same naming convention:

```
[edit]
bridge-domains {
  sales-vlan-100 { # You must use this name explicitly
    vlan-id 100;
  }
}
```

The following limitations apply to automatic bridge domain configuration:

- Only one **vlan-id-list** statement is allowed in a routing instance.
- Bridge options are not supported with the **vlan-id-list** statement.
- Only trunk interfaces are supported.
- There is no support for integrated routing and bridging (IRB).

You display the status and other parameters for automatic bridge domains configured with the **vlan-id-list** statement using the same **show l2-learning instance** command as used for individually configured bridge domains.

Related Documentation

- Ethernet Networking
- [Bulk Configuration of VLANs and Bridge Domains on page 63](#)
- [Example: Configuring VLAN Translation with a VLAN ID List on page 63](#)

CHAPTER 6

Dynamic Profiles for VLAN Interfaces and Protocols

This chapter discusses the following topics:

- [Dynamic Profiles for VPLS Pseudowires on page 67](#)
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Basic Solutions on page 68](#)
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Complex Solutions on page 72](#)

Dynamic Profiles for VPLS Pseudowires

A router often has two types of interfaces:

- Static interfaces, which are configured before the router is booted
- Dynamic interfaces, which are created after the router is booted and while it is running

A virtual private LAN service (VPLS) pseudowire interface (such as `lsi.1048576`) is dynamically created by the system. Therefore, the logical interface unit number for the VPLS pseudowire is not available in advance to configure characteristics such as virtual local area network (VLAN) identifiers and other parameters. As a result, certain VLAN manipulation features that are easily applied to static interfaces (such as `xe-`, `ge-`, and so on) are either not supported on dynamic interfaces or supported in a nonstandard method.

However, on MX Series routers, there is another configuration method that dynamic interfaces can use to determine their VLAN parameters when they are created by a running router: *dynamic profiles*. A dynamic profile is a conceptual container that includes parameters associated with a dynamic entity, parameters whose values are not known at the time the entity is configured.

A dynamic profile acts as a kind of template that enables you to create, update, or remove a configuration that includes client access (for example, interface or protocol) or service (for example, CoS) attributes. Using these profiles you can consolidate all of the common attributes of a client (and eventually a group of clients) and apply the attributes simultaneously. The router contains several predefined variables that enable dynamic association of interfaces and logical units to incoming subscriber requests. While

configuring a dynamic profile, use the **\$junos-interface-ifd-name** variable for a dynamic physical interface and the **\$junos-underlying-unit-number** variable for a dynamic logical interface (unit). When a client accesses the router, the dynamic profile configuration replaces the predefined variable with the actual interface name or unit value for the interface the client is accessing.

Dynamic profiles for VPLS are supported only on MX Series routers. For more information about dynamic profiles, see the Junos OS Subscriber Management, Release 13.1.

**Related
Documentation**

- Ethernet Networking
- Dynamic Profiles Overview
- Use Cases for Dynamic Profiles for VPLS Pseudowires
- Example: Configuring the Router to Push an Extra VLAN Tag onto Pseudowire Traffic Using Dynamic Profiles
- Example: Configuring a VPLS Pseudowire as a Trunk Interface Using Dynamic Profiles
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Basic Solutions on page 68](#)
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Complex Solutions on page 72](#)

Example: Configuring VPLS Pseudowires with Dynamic Profiles—Basic Solutions

The following limitations apply to dynamic profiles for VPLS on MX Series routers:

- The **native-vlan-id** statement is not supported.
- The **native-inner-vlan-id** statement is not supported.
- The **interface-mode access** statement option is not supported.
- The **vlan-id-range** statement is not supported.

In many cases, a configuration using dynamic profiles is more efficient than a static configuration, as shown by the examples in this topic.

- [VPLS Pseudowire Interfaces Without Dynamic Profiles on page 68](#)
- [VPLS Pseudowire Interfaces and Dynamic Profiles on page 69](#)
- [CE Routers Without Dynamic Profiles on page 70](#)
- [CE Routers and Dynamic Profiles on page 71](#)

VPLS Pseudowire Interfaces Without Dynamic Profiles

Consider the following configuration, which does not use dynamic profiles to manipulate VLAN identifiers:

```
[edit routing-instances]
green {
  instance-type vpls;
  interface ge-0/0/1.1;
```



```

interface ge-0/0/2.1;
interface ge-0/0/3.1;
vlan-tags outer 200 inner 100;
protocols vpls {
    vpls-id 10;
    neighbor 10.1.1.20;
}
{...more...}
}

[edit interfaces]
ge-0/0/1 {
    unit 0 {
        vlan-id 10;
    }
}
ge-0/0/2 {
    unit 0 {
        vlan-id 20;
    }
}
ge-0/0/3 {
    unit 0 {
        vlan-id 30;
    }
}
}

```



NOTE: This is not a complete router configuration.

With this configuration, broadcast packets inside frames arriving with VLAN identifier 10 on **ge-0/0/1** are normalized to a dual-tagged frame with an outer VLAN value of 200 and an inner VLAN value of 100. The broadcast packet and frames exiting **ge-0/0/2** or **ge-0/0/3** have the outer VLAN value stripped and the inner VLAN value swapped to 20 and 30 respectively, according to the interface configuration. However, this stripping of the outer VLAN tag and the swapping is extra work, because the frames will still egress the VPLS pseudowire in routing instance **green** with an outer VLAN tag value of 200 and an inner VLAN tag value of 100, also according to the configuration.

The same configuration can be accomplished more effectively using dynamic profiles.

VPLS Pseudowire Interfaces and Dynamic Profiles

Consider the following configuration, which uses dynamic profiles to manipulate VLAN identifiers:

```

[edit routing-instances]
green {
    instance-type vpls;
    interface ge-0/0/1.1;
    interface ge-0/0/2.1;
    interface ge-0/0/3.1;
    vlan-id 100; # Desired inner VLAN tag on the VPLS pseudowire
    protocols vpls {

```

```
vpls-id 10;
neighbor 10.1.1.20 {
    associate-profile green_vpls_pw_1; # The profile
}
}
{...more...}
}

[edit interfaces]
ge-0/0/1 {
    unit 0 {
        vlan-id 10;
    }
}
ge-0/0/2 {
    unit 0 {
        vlan-id 20;
    }
}
ge-0/0/3 {
    unit 0 {
        vlan-id 30;
    }
}

[edit dynamic-profiles]
green_vpls_pw_1 interfaces $junos-interface-ifd-name {
    unit $junos-underlying-unit-number {
        vlan-tags outer 200 inner 100;
    }
}
}
```



NOTE: This is not a complete router configuration.

With this configuration, broadcast packets inside frames arriving with VLAN identifier 10 on **ge-0/0/1** are normalized to a frame with VLAN identifier 100. The broadcast packet and frames exiting **ge-0/0/2** or **ge-0/0/3** have this VLAN value swapped to 20 and 30 respectively, according to the interface configuration. Frames egress the VPLS pseudowire in routing instance **green** with an outer VLAN tag value of 200 pushed on top of the normalized value.

CE Routers Without Dynamic Profiles

You can apply a dynamic profile to an entire VPLS configuration, not just a neighbor.

Consider the following configuration, which does not use dynamic profiles to manipulate VLAN identifiers on a customer edge (CE) router with VLAN identifier 100:

```
[edit routing-instances]
green {
    instance-type vpls;
    interface ge-0/0/1.1;
    interface ge-0/0/2.1;
```

```

interface ge-0/0/3.1;
vlan-tags outer 200 inner 100;
protocols vpls {
    vpls-id 10;
    neighbor 10.1.1.20;
}
{...more...}
}

[edit interfaces]
ge-0/0/1 {
    unit 0 {
        vlan-id 100;
    }
}
ge-0/0/2 {
    unit 0 {
        vlan-id 100;
    }
}
ge-0/0/3 {
    unit 0 {
        vlan-id 100;
    }
}
}

```



NOTE: This is not a complete router configuration.

With this configuration, broadcast packets inside frames arriving on **ge-0/0/1** are normalized to a dual-tagged frame with an outer VLAN value of 200 and an inner VLAN value of 100. The same configuration can be accomplished using dynamic profiles.

CE Routers and Dynamic Profiles

Consider the following configuration, which uses dynamic profiles at the **protocols** level:

```

[edit routing-instances]
green {
    instance-type vpls;
    interface ge-0/0/1.1;
    interface ge-0/0/2.1;
    interface ge-0/0/3.1;
    vlan-id 100; # Desired inner VLAN tag on the VPLS pseudowire
    protocols vpls {
        associate-profile green_vpls_pw_2; # The profile
        vpls-id 10;
        neighbor 10.1.1.20;
    }
    {...more...}
}

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

```

```

    unit 0 {
        vlan-id 100;
    }
}
ge-0/0/2 {
    unit 0 {
        vlan-id 100;
    }
}
ge-0/0/3 {
    unit 0 {
        vlan-id 100;
    }
}

[edit dynamic-profiles]
green_vpls_pw_2 interfaces $junos-interface-ifd-name {
    unit $junos-underlying-unit-number {
        vlan-tags outer 200 inner 100;
    }
}

```



NOTE: This is not a complete router configuration.

With this configuration, broadcast packets inside frames arriving with VLAN identifier 100 on **ge-0/0/1** are normalized to a frame with VLAN identifier 100 (in this case, they are unchanged). The broadcast packet and frames exiting **ge-0/0/2** or **ge-0/0/3** are unchanged as well, according to the interface configuration. Frames egress the VPLS pseudowire in routing instance **green** with an outer VLAN tag value of 200 pushed on top of the normalized value.

Related Documentation

- [Dynamic Profiles for Interfaces and Protocols](#)
- [Dynamic Profiles for VPLS Pseudowires on page 67](#)
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Complex Solutions on page 72](#)

Example: Configuring VPLS Pseudowires with Dynamic Profiles—Complex Solutions

Dynamic profiles for VPLS pseudowires can be helpful in a variety of VLAN configurations. This section explores some of these situations through examples.



NOTE: These examples are not complete router configurations.

All of the examples in this section address the same basic topology. A routing instance **blue** uses a trunk bridge to connect different departments in an organization, each with

their own VLANs, at two different sites. The organization uses a BGP-based VPLS with a virtual switch to accomplish this.

- [Configuration of Routing Instance and Interfaces Without Dynamic Profiles on page 73](#)
- [Configuration of Routing Instance and Interfaces Using Dynamic Profiles on page 74](#)
- [Configuration of Tag Translation Using Dynamic Profiles on page 76](#)

Configuration of Routing Instance and Interfaces Without Dynamic Profiles

The basic configuration of routing instance and interfaces without dynamic profiles follows:

```
[edit routing-instance blue]
instance-type virtual-switch;
route-distinguisher 10.1.1.10:1;
vrf-target target:1000:1;
interface ge-3/0/0; # The trunk interface
bridge-domains {
  sales {
    vlan-id 10;
    interface ge-0/0/0.1;
    ... # Other interfaces and statements for Sales
  }
  engineering {
    vlan-id 20;
    interface ge-1/0/2.0;
    ... # Other interfaces and statements for Engineering
  }
  accounting {
    vlan-id 30;
    interface ge-2/0/3.0;
    ... # Other interfaces and statements for Accounting
  }
  others {
    vlan-id-list [ 40 50 ]; # Other departments
  }
}
protocols vpls {
  site-range 10;
  site sample-site-1 {
    site-identifier 1;
  }
}
... # Other statements for instance Blue

[edit interfaces]
ge-0/0/1 {
  unit 0 {
    vlan-id 100;
  }
}
ge-3/0/0 {
  unit 0 {
    family bridge {
```

```
        interface-mode trunk; # This is the trunk
        vlan-id-list [ 10 20 30 40 50 ];
    }
}
... # More interface statements
```

This configuration switches the departmental VLAN traffic (sales, engineering, etc.) bridge domains over the VPLS pseudowire trunk connecting to the other site.

Configuration of Routing Instance and Interfaces Using Dynamic Profiles

Here is how dynamic profiles can be applied to this basic configuration.

First, consider the requirement to push an outer VLAN tag value of 200 onto the VPLS pseudowire frames on egress. Dynamic profiles easily satisfy this requirement.

```
[edit routing-instance green]
instance-type virtual-switch;
... # Other routing instance statements
protocols vpls {
    site-range 10;
    site sample-site-1 {
        site-identifier 1;
    }
    associate-profile green_vpls_pw_1; # Apply profile here
}
... # Other routing instance statements

[edit dynamic-profiles]
green_vpls_pw_1 interfaces $junos-interface-ifd-name {
    unit $junos-underlying-unit-number {
        vlan-id 200; # This is the outer tag
        family bridge {
            interface-mode trunk;
            inner-vlan-id-list [ 10 20 30 40 50 ];
        }
    }
}
```



NOTE: This is not a complete router configuration.

With the dynamic profile, a packet in a frame arriving on an interface is classified as belonging to one of the bridge domains (VLANs 10–50). At the egress of the trunk VPLS pseudowire, the outer VLAN tag 200 is pushed onto the frame. At the ingress of the pseudowire at the remote location, the outer VLAN tag 200 is removed and the frame is delivered to the appropriate bridge domain.

But what if the packets associated with the Accounting VLAN are not to be forwarding to the remote site? Dynamic profiles are useful here as well.

This configuration keeps the Accounting frames from reaching the remote site.

```
[edit routing-instances green]
```

```

instance-type virtual-switch;
... # Other routing instance statements
protocols vpls {
  site-range 10;
  site sample-site-1 {
    site-identifier 1;
  }
  associate-profile green_vpls_pw_2; # Apply profile here
}
... # Other routing instance statements

[edit dynamic-profiles]
green_vpls_pw_2 interfaces $junos-interface-ifd-name {
  unit $junos-underlying-unit-number {
    family bridge {
      interface-mode trunk;
      inner-vlan-id-list [ 10 20 40 50 ]; # Removed Accounting VLAN 30
    }
  }
}

```



NOTE: This is not a complete router configuration.

In this case, frames arriving on the interfaces are classified according to their bridge domains and switched, if necessary, to the VPLS pseudowire trunk, except for Engineering frames. Engineering frames (VLAN 30) are only switched within the interfaces listed within bridge domain **accounting** and any statically configured trunk interfaces and are prevented from crossing the VPLS pseudowire due to the absence of VLAN 30 on the trunk.

We can combine the two examples and use dynamic profiles to forward the frames (other than **accounting** frames) to the remote site with an out tag of 200.

This configuration keeps the Accounting frames from reaching the remote site and pushes an outer tag of 200 on VPLS pseudowire traffic.

```

[edit routing-instances green]
instance-type virtual-switch;
... # Other routing instance statements
protocols vpls {
  site-range 10;
  site sample-site-1 {
    site-identifier 1;
  }
  associate-profile green_vpls_pw_3; # Apply profile here
}
... # Other routing instance statements

[edit dynamic-profiles]
green_vpls_pw_3 interfaces $junos-interface-ifd-name {
  unit $junos-underlying-unit-number {
    vlan-id 200; # This is the outer tag
    family bridge {

```

```

        interface-mode trunk;
        inner-vlan-id-list [ 10 20 40 50 ]; # Removed Accounting VLAN 30
    }
}

```



NOTE: This is not a complete router configuration.

In this case, frames arriving on the interfaces are classified according to their bridge domains and switched, if necessary, to the VPLS pseudowire trunk with an outer VLAN tag of 200, except for Engineering frames. Engineering frames (VLAN 30) are only switched within the interfaces listed within bridge domain **accounting** and any statically configured trunk interfaces and are prevented from crossing the VPLS pseudowire due to the absence of VLAN 30 on the trunk.

Configuration of Tag Translation Using Dynamic Profiles

Consider a final case where the bridge domain VLANs need translation at the VPLS pseudowire trunk interface. In this case, **sales** (VLAN 10) is mapped to VLAN 110 and **engineering** (VLAN 20) is mapped to VLAN 120.

This configuration adds tag translation to the VPLS pseudowire traffic.

```

[edit routing-instances green]
instance-type virtual-switch;
... # Other routing instance statements
protocols vpls {
    site-range 10;
    site sample-site-1 {
        site-identifier 1;
    }
    associate-profile green_vpls_pw_4; # Apply profile here
}
... # Other routing instance statements

```

```

[edit dynamic-profiles]
green_vpls_pw_4 interfaces $junos-interface-ifd-name {
    unit $junos-underlying-unit-number {
        family bridge {
            interface-mode trunk;
            vlan-id-list [ 10 20 30 40 50 ]; # All VLANs
            vlan-rewrite translate 110 10; # Sales VLAN
            vlan-rewrite translate 120 20; # Engineering VLAN
        }
    }
}

```



NOTE: This is not a complete router configuration.

This translates the **sales** and **engineering** VLAN tags exiting the VPLS pseudowire accordingly. At the ingress of the VPLS pseudowire, VLANs 110 and 120 are translated back to 10 and 20, respectively.

**Related
Documentation**

- [Dynamic Profiles for Interfaces and Protocols](#)
- [Dynamic Profiles for VPLS Pseudowires on page 67](#)
- [Example: Configuring VPLS Pseudowires with Dynamic Profiles—Basic Solutions on page 68](#)

CHAPTER 7

MX Series Router as a DHCP Relay Agent

This chapter discusses the following topics:

- [MX Series Router as a Layer 2 DHCP Relay Agent on page 79](#)
- [Example: Configuring DHCP Relay in a Bridge Domain VLAN Environment on page 80](#)
- [Example: Configuring DHCP Relay in a VPLS Routing Instance Environment on page 81](#)

MX Series Router as a Layer 2 DHCP Relay Agent

The Dynamic Host Configuration Protocol (DHCP) is used by a DHCP client (host) to determine Layer 3 information (such as an IP address) from a DHCP server. DHCP uses the client's MAC (Layer 2) address to query the server. A router can be used as a DHCP relay agent to pass the query on to a server while the router appears to reply to the client. You can configure a Juniper Networks MX Series Ethernet Services Router to act as a DHCP relay agent. The MX Series router configuration at Layer 2 accesses the Layer 3 information with DHCP snooping.

DHCP servers and relay agents have a level of trust in the MAC addresses used in DHCP client queries. A hacker can spoof invalid MAC addresses and overwhelm the server or relay agent with flooded traffic. Or the hacker can try to determine other information, such as the IP address range used by devices on the network. The DHCP process should only trust MAC addresses that are valid for a particular network.

You can configure the MX Series router to use MAC addresses obtained by the Layer 2 address learning process to control the flooding of DHCP packets.

Several restrictions apply to DHCP configuration on the MX Series routers:

- All statements referring to “option 82” (including circuit information in DHCP relay messages) are not supported on the MX Series routers.
- This feature works for static IP/MAC bindings on the MX Series routers.
- The DHCP snooping database table is not restored after a Routing Engine reboot.
- The DHCP Discover message is not flooded to the DHCP server when broadband service aggregator (BSA) and broadband service router (BSR) are provisioned on the same switch.

For more information on configuring DHCP, see the Junos OS Subscriber Management, Release 13.1.

**Related
Documentation**

- DHCP Relay Agent
- [Example: Configuring DHCP Relay in a Bridge Domain VLAN Environment on page 80](#)
- [Example: Configuring DHCP Relay in a VPLS Routing Instance Environment on page 81](#)

Example: Configuring DHCP Relay in a Bridge Domain VLAN Environment

The following example configures DHCP relay in a VPLS environment to trust only the MAC addresses learned on the listed interfaces.



NOTE: This is not a complete router configuration.

```
[edit]
routing-instances {
  classic-vpls {
    instance-type vpls;
    interface ge-1/1/1.0;
    interface ge-1/1/2.0;
    interface ge-1/1/3.0;
    interface ge-1/1/4.0;
    interface ge-1/1/5.0;
    vlan-id 20;
    forwarding-options {
      dhcp-relay { # Here is where DHCP is configured.
        group vlan-20-bridge {
          interface ge-1/1/1.0;
          interface ge-1/1/2.0;
          interface ge-1/1/3.0;
          interface ge-1/1/4.0;
          interface ge-1/1/5.0;
        }
      }
    }
    protocol vpls {
      site-id 567;
      # Other VPLS configuration statements...
    }
  }
}
```

Only MAC addresses learned on the five listed Gigabit Ethernet interfaces will be trusted for DHCP relay purposes.

For more information on configuring DHCP, see the Junos OS Subscriber Management, Release 13.1.

**Related
Documentation**

- DHCP Relay Agent
- [MX Series Router as a Layer 2 DHCP Relay Agent on page 79](#)

- [Example: Configuring DHCP Relay in a VPLS Routing Instance Environment on page 81](#)

Example: Configuring DHCP Relay in a VPLS Routing Instance Environment

The following example configures DHCP relay in a bridge domain (VLAN) environment. The MX Series router will trust only the MAC addresses learned on the listed interfaces.



NOTE: This is not a complete router configuration.

The router has three interfaces: two interfaces (**ge-2/2/4** and **ge-2/2/6**) using VLAN 100 for the DHCP clients, and one (**xe-9/2/0**) leading to the DHCP server. The router performs the DHCP snooping (relay) function.

Configure the Interfaces

```
[edit interfaces]
ge-2/2/4 {
  encapsulation ethernet-bridge;
  unit 0 {
    family bridge {
      interface-mode access;
      vlan-id 100;
    }
  }
}
ge-2/2/6 {
  encapsulation ethernet-bridge;
  unit 0 {
    family bridge {
      interface-mode access;
      vlan-id 100;
    }
  }
}
xe-9/2/0 {
  unit 0 {
    family bridge {
      interface-mode access;
      vlan-id 100;
    }
  }
}
```

Configure the Routing Instance (Virtual Switch)

```
[edit routing-instances]
vsl {
  instance-type virtual-switch;
  interface ge-2/2/4.0;
  interface ge-2/2/6.0;
  interface xe-9/2/0;
  bridge-domains {
    bd1 {
      domain-type bridge;
      vlan-id 100;
      forwarding-options {
```

```
        dhcp-relay { # DHCP snooping
            group hdhcp {
                interface ge-2/2/4.0;
                interface ge-2/2/6.0;
            }
        }
    }
}
```

You verify your configuration by using two related commands:

- **show dhcp relay binding routing-instance vs1 bridge-domains bd1**
- **show dhcp relay binding routing-instance vs1 bridge-domains bd1 detail**

```
user@router1> show dhcp relay binding routing-instance vs1 bridge-domains bd1
2 clients, (2 bound, 0 selecting, 0 renewing, 0 rebinding)
IP address      Hardware address  Type    Lease expires at
192.168.1.1     00:00:00:42:a8:e3 active    2008-12-12 15:56:04 PST
192.168.1.2     00:00:00:42:a8:e4 active    2008-12-12 15:56:10 PST
```

```
user@router1> show dhcp relay binding routing-instance vs1 bridge-domains bd1 detail
2 clients, (2 bound, 0 selecting, 0 renewing, 0 rebinding)
Clients bindings information:
IP address      : 192.168.1.1
Hardware address : 00:00:00:42:a8:e3
Type            : active
Lease expires at : 2008-12-12 15:56:04 PST
State           : bound
interface       : ge-2/2/6.0
IP address      : 192.168.1.2
Hardware address : 00:00:00:42:a8:e4
Type            : active
Lease expires at : 2008-12-12 15:56:10 PST
State           : bound
interface       : ge-2/2/4.0
```

**Related
Documentation**

- [DHCP Relay Agent](#)
- [MX Series Router as a Layer 2 DHCP Relay Agent on page 79](#)
- [Example: Configuring DHCP Relay in a Bridge Domain VLAN Environment on page 80](#)

CHAPTER 8

MX Series Router in an ATM Ethernet Interworking Function

This chapter discusses the following topics:

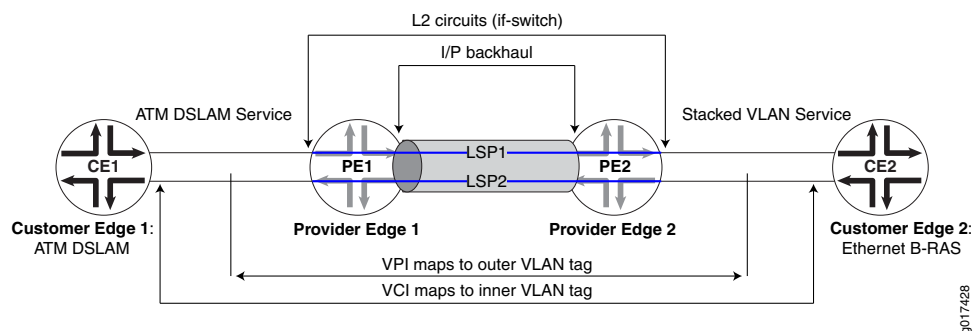
- [MX Series Router ATM Ethernet Interworking Function on page 83](#)
- [Example: Configuring MX Series Router ATM Ethernet Interworking on page 85](#)

MX Series Router ATM Ethernet Interworking Function

You can configure an MX Series router as part of an ATM Ethernet interworking function (IWF) scenario mapping outer and inner VLAN tags to ATM Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI).

The ATM Ethernet interworking scenario is shown in [Figure 10 on page 83](#). The MX Series router is configured as the Provider Edge 2 (PE2) router in the figure to support the ATM Ethernet IWF. Ethernet is the only transport type supported.

Figure 10: ATM Ethernet VLAN Interworking



The PE1 router translates between ATM and Ethernet VLANs. Only an M Series router can function as the PE1 router.

The PE1 router translates between the ATM VPI and VCI and Ethernet VLAN tags as follows:

- ATM VPI to and from outer VLAN tag of the Ethernet frame
- ATM VCI to and from inner VLAN tag of the Ethernet frame

Because of the translation, the flow of packets and frames between PE1 (the M Series router) and PE2 (the MX series router) routers is not symmetrical, as is shown in [Figure 11 on page 84](#).

Figure 11: ATM Ethernet VLAN Interworking Packet Structure

1. PE1 → PE2

L3	Ethertype	SA	DA	Inner VLAN	MPLS	Ethernet
----	-----------	----	----	------------	------	----------

“8 bytes” is an ATM cookie added by an M Series ATM pic.
The first 2 bytes of this ATM cookie is inner VLAN.

2. PE2 → PE1

L3	Ethertype	SA	DA	Inner VLAN	MPLS	Ethernet
----	-----------	----	----	------------	------	----------

g017429

For PE1 to PE2 traffic, the 8 bytes following the MPLS header is an ATM cookie added by the M Series ATM PIC. The first two bytes are the inner VLAN tag, which is why the field extends to the right of the figure.

The traffic between PE2 and CE2 is a normal flow of stacked Ethernet frames.

You can also configure a CCC with remote interface switch or Layer 2 circuit over Aggregated Ethernet on the MX Series router (PE2). When CCC is configured for Aggregated Ethernet, the flow of packets is as shown in [Figure 12 on page 84](#).

Figure 12: CCC-to-Stacked VLAN Translation

1. CCC to stacked-vlan

L3	Ethertype	SA	DA	Inner VLAN	MPLS	Ethernet
L3	Ethertype	Ivlan	Ovlan	SA	DA	

2. Stacked-vlan to CCC

L3	Ethertype	Ivlan	Ovlan	SA	DA	
L3	Ethertype	SA	DA	Inner VLAN	MPLS	Ethernet

g017430

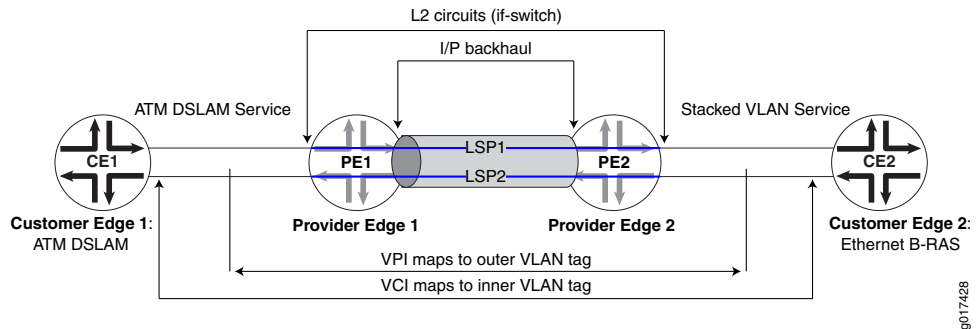
Related Documentation

- [ATM Ethernet Interworking](#)
- [Configuring MX Series Router ATM Ethernet Interworking on page 85](#)

Example: Configuring MX Series Router ATM Ethernet Interworking

Consider the router topology shown in [Figure 13 on page 85](#). The MX Series router is configured as the Router PE2 (the provider edge 2 router) in the figure to support the ATM Ethernet IWF.

Figure 13: ATM Ethernet VLAN Interworking



The relevant router interfaces are as follows:

- On Router CE1, the CE1–PE1 interface is **at-2/0/0** with IP address **30.1.1.1/24**.
- On Router PE1:
 - The PE1–CE1 interface is **at-2/0/1** with no IP address required.
 - The PE1–PE2 interface is **ge-5/0/0** with IP address **20.1.1.1/24**.
- On Router PE2:
 - The PE2–CE2 interface is **ge-0/2/0** with no IP address required.
 - The PE2–PE1 interface is **ge-0/2/8** with IP address **20.1.1.10/24**.
- On Router CE2 interface is **ge-0/0/0** with IP address **30.1.1.10/24**.



NOTE: The example configurations in this section are not complete configurations.

Consider the following example MX Series router configurations.

- [Configuring Router PE2 with a Layer 2 Circuit on page 86](#)
- [Configuring Router PE2 with a Layer 2 Circuit over Aggregated Ethernet on page 88](#)
- [Configuring Router PE2 with a Remote Interface Switch on page 91](#)
- [Configuring Router PE2 with a Remote Interface Switch over Aggregated Ethernet on page 94](#)

Configuring Router PE2 with a Layer 2 Circuit

Router CE1 Configuration The configuration of the Layer 2 circuit is based on LDP-signaled MPLS connections. Configure Ethernet over ATM on the ATM interface.

```
[edit]
interfaces {
  at-2/0/0 {
    encapsulation ethernet-over-atm;
    atm-options {
      vpi 100;
    }
    unit 0 {
      vci 100.34;
      family inet {
        address 30.1.1.1/24;
      }
    }
  }
}
```

Router PE1 Configuration Configure the Layer 2 circuit.

```
[edit]
interfaces {
  at-2/0/1 {
    atm-options {
      vpi 100;
    }
    unit 0 {
      encapsulation vlan-vci-ccc;
      vpi 100;
      vci-range 32 63;
    }
  }
  ge-5/0/0 {
    unit 0 {
      family inet {
        address 20.1.1.1/24;
      }
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface ge-5/0/0.0;
  }
  ospf {
    area 0.0.0.0 {
      interface ge-5/0/0.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

```

    }
  }
  ldp {
    interface all;
  }
  l2circuit {
    neighbor 10.255.171.14 {
      interface at-2/0/1.0 {
        virtual-circuit-id 100;
      }
    }
  }
}

```

Router PE2 Configuration Configure the Layer 2 circuit on the MX Series router.

```

[edit]
interfaces {
  ge-0/2/0 {
    vlan-vci-tagging;
    encapsulation vlan-vci-ccc;
    unit 0 {
      vlan-id 100;
      inner-vlan-id-range start 32 end 63;
    }
  }
  ge-0/2/8 {
    unit 0 {
      family inet {
        address 20.1.1.10/24;
      }
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface ge-0/2/8.0;
  }
  ospf {
    area 0.0.0.0 {
      interface ge-0/2/8.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
ldp {
  interface all;
}
l2circuit {
  neighbor 10.255.171.45 {
    interface ge-0/2/0.0 {
      virtual-circuit-id 100;
    }
  }
}

```

```
}
}
```

Router CE2 Configure the dual- tagged Ethernet interface.

Configuration

```
[edit]
interfaces {
  ge-0/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      vlan-tags outer 100 inner 34;
      family inet {
        address
          30.1.1.10/24;
      }
    }
  }
}
```

You verify your configuration on the MX Series router with the **show l2circuit connections** command:

```
user@PE2>show l2circuit connections
```

Layer-2 Circuit Connections:

Legend for connection status (St)

EI -- encapsulation invalid	NP -- interface h/w not present
MM -- mtu mismatch	Dn -- down
EM -- encapsulation mismatch	VC-Dn -- Virtual circuit Down
CM -- control-word mismatch	Up -- operational
VM -- vlan id mismatch	CF -- Call admission control failure
OL -- no outgoing label	IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC	TM -- TDM misconfiguration
BK -- Backup Connection	ST -- Standby Connection
CB -- rcvd cell-bundle size bad	XX -- unknown
SP -- Static Pseudowire	

Legend for interface status

Up -- operational
Dn -- down

Neighbor: 10.255.171.45

Interface	Type	St	Time last up	# Up trans
ge-0/2/0.0(vc 100)	rmt	Up	May 12 13:01:50 2009	1

Remote PE: 10.255.171.45, Negotiated control-word: Yes (Null)

Incoming label: 299824, Outgoing label: 299776

Local interface: ge-0/2/0.0, Status: Up, Encapsulation: VLAN Q-in-Q and

VCI Interworking

Configuring Router PE2 with a Layer 2 Circuit over Aggregated Ethernet

Router CE1 The configuration of the Layer 2 circuit is based on LDP-signaled MPLS connections.

Configuration

Configure Ethernet over ATM on the ATM interface.

```
[edit]
interfaces {
  at-2/0/0 {
    encapsulation ethernet-over-atm;
```

```

    atm-options {
        vpi 100;
    }
    unit 0 {
        vci 100.34;
        family inet {
            address 30.1.1.1/24;
        }
    }
}

```

Router PE1 Configuration

Configure the Layer 2 circuit.

```

[edit]
interfaces {
    at-2/0/1 {
        atm-options {
            vpi 100;
        }
        unit 0 {
            encapsulation vlan-vci-ccc;
            vpi 100;
            vci-range 32 63;
        }
    }
    ge-5/0/0 {
        unit 0 {
            family inet {
                address 20.1.1.1/24;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        interface ge-5/0/0.0;
    }
    ospf {
        area 0.0.0.0 {
            interface ge-5/0/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
ldp {
    interface all;
}
l2circuit {
    neighbor 10.255.171.14 {
        interface at-2/0/1.0 {
            virtual-circuit-id 100;
        }
    }
}

```

```
}  
}
```

**Router PE2
Configuration**

Configure the Layer 2 circuit over aggregated Ethernet on the MX Series router.

```
[edit]  
chassis {  
  aggregated-devices {  
    ethernet {  
      device-count 1;  
    }  
  }  
}  
interfaces {  
  ge-0/2/0 {  
    gigether-options {  
      802.3ad ae0;  
    }  
  }  
  ge-0/2/8 {  
    unit 0 {  
      family inet {  
        address 20.1.1.10/24;  
      }  
      family mpls;  
    }  
  }  
  ae0 {  
    vlan-vci-tagging;  
    encapsulation vlan-vci-ccc;  
    unit 0 {  
      vlan-id 100;  
      inner-vlan-id-range start 32 end 63;  
    }  
  }  
}  
protocols {  
  mpls {  
    interface ge-0/2/8.0;  
  }  
  ospf {  
    area 0.0.0.0 {  
      interface ge-0/2/8.0;  
      interface lo0.0 {  
        passive;  
      }  
    }  
  }  
  ldp {  
    interface all;  
  }  
  l2circuit {  
    neighbor 10.255.171.45 {  
      interface ae0.0 {  
        virtual-circuit-id 100;  
      }  
    }  
  }  
}
```

```

    }
  }
}

```

Router CE2 Configuration Configure the dual-tagged Ethernet interface.

```

[edit]
interfaces {
  ge-0/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      vlan-tags outer 100 inner 34;
      family inet {
        address
          30.1.1.10/24;
      }
    }
  }
}

```

You verify your configuration on the MX Series router with the **show l2circuit connections** command:

```

user@PE2>show l2circuit connections
Layer-2 Circuit Connections:

```

Legend for connection status (St)

EI -- encapsulation invalid	NP -- interface h/w not present
MM -- mtu mismatch	Dn -- down
EM -- encapsulation mismatch	VC-Dn -- Virtual circuit Down
CM -- control-word mismatch	Up -- operational
VM -- vlan id mismatch	CF -- Call admission control failure
OL -- no outgoing label	IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC	TM -- TDM misconfiguration
BK -- Backup Connection	ST -- Standby Connection
CB -- rcvd cell-bundle size bad	XX -- unknown
SP -- Static Pseudowire	

Legend for interface status

Up -- operational
Dn -- down

Neighbor: 10.255.171.45

Interface	Type	St	Time last up	# Up trans
ae0.0(vc 100)	rmt	Up	May 12 14:48:58 2009	2
Remote PE: 10.255.171.45, Negotiated control-word: Yes (Null)				
Incoming label: 299872, Outgoing label: 299824				
Local interface: ae0.0, Status: Up, Encapsulation:VLAN Q-in-Q and VCI				

Interworking

Configuring Router PE2 with a Remote Interface Switch

Router CE1 Configuration The configuration of the remote interface switch is based on RSVP-signaled MPLS connections.

Configure Ethernet over ATM on the ATM interface.

```

[edit]
interfaces {

```

```
at-2/0/0 {
  encapsulation ethernet-over-atm;
  atm-options {
    vpi 100;
  }
  unit 0 {
    vci 100.34;
    family inet {
      address 30.1.1.1/24;
    }
  }
}
```

Router PE1 Configuration Configure the remote interface switch.

```
[edit]
interfaces {
  at-2/0/1 {
    atm-options {
      vpi 100;
    }
    unit 0 {
      encapsulation vlan-vci-ccc;
      vpi 100;
      vci-range start 32 end 63;
    }
  }
  ge-5/0/0 {
    unit 0 {
      family inet {
        address 20.1.1.1/24;
      }
      family iso;
      family mpls;
    }
  }
}
protocols {
  rsvp {
    interface ge-5/0/0.0;
  }
  mpls {
    label-switched-path lsp1-2 {
      from 10.255.171.45;
      to 10.255.171.14;
    }
    label-switched-path lsp2-1 {
      from 10.255.171.14;
      to 10.255.171.45;
    }
    interface ge-5/0/0.0;
  }
  isis {
    interface ge-5/0/0.0;
  }
}
```



```

connections {
  remote-interface-switch rws1 {
    interface at-2/0/1.0;
    transmit-lsp lsp1-2;
    receive-lsp lsp2-1;
  }
}

```

Router PE2 Configuration Configure the remote interface switch on the MX Series router.

```

[edit]
interfaces {
  ge-0/2/0 {
    vlan-vci-tagging;
    encapsulation vlan-vci-ccc;
    unit 0 {
      vlan-id 100;
      inner-vlan-id-range start 32 end 63;
    }
  }
  ge-0/2/8 {
    unit 0 {
      family inet {
        address 20.1.1.10/24;
      }
      family iso;
      family mpls;
    }
  }
}
protocols {
  rsvp {
    interface ge-0/2/8.0;
  }
  mpls {
    label-switched-path lsp2-1 {
      from 10.255.171.14;
      to 10.255.171.45;
    }
    label-switched-path lsp1-2 {
      from 10.255.171.45;
      to 10.255.171.14;
    }
    interface ge-0/2/8.0;
  }
  isis {
    interface ge-0/2/8.0;
  }
  connections {
    remote-interface-switch rws1 {
      interface ge-0/2/0.0;
      transmit-lsp lsp2-1;
      receive-lsp lsp1-2;
    }
  }
}

```

```
}

```

Router CE2 Configuration Configure the dual-tagged Ethernet interface.

```
[edit]
interfaces {
  ge-0/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      vlan-tags outer 100 inner 34;
      family inet {
        address
          30.1.1.10/24;
      }
    }
  }
}
```

You verify your configuration on the MX Series router with the **show connections** command:

```
user@PE2>show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St)
UN -- uninitialized
NP -- not present
WE -- wrong encapsulation
DS -- disabled
Dn -- down
-> -- only outbound conn is up
<- -- only inbound conn is up
Up -- operational
RmtDn -- remote CCC down
Restart -- restarting

Legend for connection types
if-sw: interface switching
rmt-if: remote interface switching
lsp-sw: LSP switching
tx-p2mp-sw: transmit P2MP switching
rx-p2mp-sw: receive P2MP switching

Legend for circuit types
intf -- interface
tlsp -- transmit LSP
rlsp -- receive LSP
```

Connection/Circuit	Type	St	Time last up	# Up
trans				
rws1	rmt-if	Up	May 13 11:25:40	
1				
ge-0/2/0.0	intf	Up		
lsp2-1	tlsp	Up		
lsp1-2	rlsp	Up		

Configuring Router PE2 with a Remote Interface Switch over Aggregated Ethernet

Router CE1 Configuration The configuration of the remote interface switch is based on RSVP-signaled MPLS connections.

Configure Ethernet over ATM on the ATM interface.

```
[edit]
interfaces {
  at-2/0/0 {
    encapsulation ethernet-over-atm;
    atm-options {
      vpi 100;
    }
  }
}
```

```

        unit 0 {
            vci 100.34;
            family inet {
                address 30.1.1.1/24;
            }
        }
    }
}

```

Router PE1 Configuration Configure the remote interface switch.

```

[edit]
interfaces {
    at-2/0/1 {
        atm-options {
            vpi 100;
        }
        unit 0 {
            encapsulation vlan-vci-ccc;
            vpi 100;
            vci-range 32 end 63;
        }
    }
    ge-5/0/0 {
        unit 0 {
            family inet {
                address 20.1.1.1/24;
            }
            family mpls;
        }
    }
}
protocols {
    rsvp {
        interface ge-5/0/0.0;
    }
    mpls {
        label-switched-path lsp1-2 {
            from 10.255.171.45;
            to 10.255.171.14;
        }
        label-switched-path lsp2-1 {
            from 10.255.171.14;
            to 10.255.171.45;
        }
        interface ge-5/0/0.0;
    }
    isis {
        interface ge-5/0/0.0;
    }
}
connections {
    remote-interface-switch rws1 {
        interface at-2/0/1.0;
        transmit-lsp lsp1-2;
        receive-lsp lsp2-1;
    }
}

```

```

    }
  }
}

```

Router PE2 Configuration

Configure the remote interface switch over aggregated Ethernet on the MX Series router.

```

[edit]
chassis {
  aggregated-devices {
    ethernet {
      device-count 1;
    }
  }
}
interfaces {
  ge-0/2/0 {
    gigether-options {
      802.3ad ae0;
    }
  }
  ge-0/2/8 {
    unit 0 {
      family inet {
        address 20.1.1.10/24;
      }
      family iso;
      family mpls;
    }
  }
  ae0 {
    vlan-vci-tagging;
    encapsulation vlan-vci-ccc;
    unit 0 {
      vlan-id 100;
      inner-vlan-id-range start 32 end 63;
    }
  }
}
protocols {
  rsvp {
    interface ge-0/2/8.0;
  }
  mpls {
    label-switched-path lsp2-1 {
      from 10.255.171.14;
      to 10.255.171.45;
    }
    label-switched-path lsp1-2 {
      from 10.255.171.45;
      to 10.255.171.14;
    }
    interface ge-0/2/08.0;
  }
  isis {
    interface ge-02/08.0; {
  }
}

```

```

connections {
  remote-interface-switch rws1 {
    interface ae0.0 {
      transmit-lsp- lsp-lsp2-1;
      receive-lsp lsp1-2;
    }
  }
}

```

Router CE2 Configuration Configure the dual-tagged Ethernet Interface.

```

[edit]
interfaces {
  ge-0/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      vlan-tags outer 100 inner 34;
      family inet {
        address
          30.1.1.10/24;
      }
    }
  }
}

```

You verify your configuration on the MX Series router with the **show connections** command:

```

user@PE2>show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St)
UN -- uninitialized
NP -- not present
WE -- wrong encapsulation
DS -- disabled
Dn -- down
-> -- only outbound conn is up
<- -- only inbound conn is up
Up -- operational
RmtDn -- remote CCC down
Restart -- restarting

Legend for connection types
if-sw: interface switching
rmt-if: remote interface switching
lsp-sw: LSP switching
tx-p2mp-sw: transmit P2MP switching
rx-p2mp-sw: receive P2MP switching

Legend for circuit types
intf -- interface
tlsp -- transmit LSP
rlsp -- receive LSP

```

Connection/Circuit	Type	St	Time last up	# Up
trans				
rws1	rmt-if	RmtDn	May 12 15:34:46	
1				
ae0.0	intf	Up		
lsp2-1	tlsp	Up		
lsp1-2	rlsp	Up		

Related Documentation

- ATM Ethernet Interworking
- [MX Series Router ATM Ethernet Interworking Function on page 83](#)

PART 3

Ethernet Filtering, Monitoring, and Fault Management Solutions for MX Series Routers

- [Layer 2 Firewall Filters on page 101](#)
- [IEEE 802.1ag OAM Connectivity-Fault Management on page 109](#)
- [ITU-T Y.1731 Ethernet Frame Delay Measurements on page 125](#)
- [IEEE 802.3ah OAM Link-Fault Management on page 143](#)
- [Ethernet Ring Protection on page 151](#)

CHAPTER 9

Layer 2 Firewall Filters

This chapter discusses the following topics:

- [Firewall Filters for Bridge Domains and VPLS Instances on page 101](#)
- [Example: Configuring Policing and Marking of Traffic Entering a VPLS Core on page 102](#)
- [Example: Configuring Filtering of Frames by MAC Address on page 104](#)
- [Example: Configuring Filtering of Frames by IEEE 802.1p Bits on page 106](#)
- [Example: Configuring Filtering of Frames by Packet Loss Priority on page 107](#)

Firewall Filters for Bridge Domains and VPLS Instances

Juniper Networks MX Series 3D Universal Edge 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 a Layer 2 environment to control traffic.

MX Series router firewall filters can be applied to:

- Input interfaces
- Output interfaces
- Input to the Layer 2 forwarding table



NOTE: Broadcast, unicast unknown, and multicast (BUM) traffic is not affected by input and output policies. BUM traffic can only be filtered by forwarding table policies.

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 Routing Policy Configuration Guide.



NOTE: If the chassis is running in Enhanced IP mode, a single shared filter instance is created for a filter applied across bridge domains. However, if the chassis is not running in Enhanced IP mode, then separate filter instances are created for each bridge domain that the filter is applied to.

Related Documentation

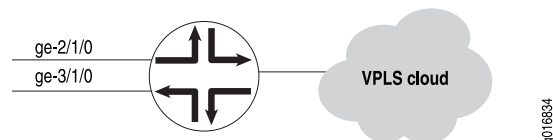
- Layer 2 Firewall Filters
- [Example: Configuring Policing and Marking of Traffic Entering a VPLS Core on page 102](#)
- [Example: Configuring Filtering of Frames by MAC Address on page 104](#)
- [Example: Configuring Filtering of Frames by IEEE 802.1p Bits on page 106](#)
- [Example: Configuring Filtering of Frames by Packet Loss Priority on page 107](#)

Example: Configuring Policing and Marking of 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 OS Class of Service Configuration Guide.

The position of the router is shown in [Figure 14 on page 102](#).

Figure 14: 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.



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

To configure policing and marking of traffic entering a VPLS core:

1. Configure **policer bcast-unknown-unicast-non-ip-mcast-policer**, a firewall policer 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;
}
```

2. Configure **three-color-policer ip-multicast-traffic-policer**, a three-color policer 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;
  }
}
```

3. Configure **customer-1**, a firewall filter that uses the two policers to limit and mark customer traffic. The first term marks the IP multicast traffic based on the 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 bcst-unknown-unicast-non-ip-mcast-policer;
}
}
}

```

4. Apply the firewall filter as an input filter to the customer interface at **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;
            }
        }
    }
}

```

Related Documentation

- [Layer 2 Firewall Filters](#)
- [Firewall Filters for Bridge Domains and VPLS Instances on page 101](#)
- [Example: Configuring Filtering of Frames by MAC Address on page 104](#)
- [Example: Configuring Filtering of Frames by IEEE 802.1p Bits on page 106](#)
- [Example: Configuring Filtering of Frames by Packet Loss Priority on page 107](#)

Example: Configuring Filtering of 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 Routing Policy Configuration Guide. The filter is applied to the VLAN configured as **vlan100200** as an input filter on Router 1.



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

To configure filtering of frames by MAC address:

1. Configure **evil-mac-address**, 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
    }
  }
}
```

2. Apply **evil-mac-address** 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;
        }
      }
    }
  }
}
```

Related Documentation

- Layer 2 Firewall Filters
- [Firewall Filters for Bridge Domains and VPLS Instances on page 101](#)
- [Example: Configuring Policing and Marking of Traffic Entering a VPLS Core on page 102](#)
- [Example: Configuring Filtering of Frames by IEEE 802.1p Bits on page 106](#)
- [Example: Configuring Filtering of Frames by Packet Loss Priority on page 107](#)

Example: Configuring Filtering of Frames by IEEE 802.1p Bits

For the **bridge** and **vpls** protocol families only, MX Series router firewall filters can be configured to provide matching on IEEE 802.1p priority bits in packets with VLAN tagging:

- To configure a firewall filter term that includes matching on IEEE 802.1p learned VLAN priority (in the outer VLAN tag), use the **learn-vlan-1p-priority** or **learn-vlan-1p-priority-except** match condition.
- To configure a firewall filter term that includes matching on IEEE 802.1p user priority (in the inner VLAN tag), use the **user-vlan-1p-priority** or **user-vlan-1p-priority-except** match condition.

For more detailed information about configuring firewall filters and configuring filter match conditions for Layer 2 bridging traffic on the MX Series routers, see the Routing Policy Configuration Guide.



NOTE: Layer 2 bridging is supported only on the MX Series routers. For more information about how to configure Layer 2 bridging, see the Routing Policy Configuration Guide, the Junos OS Routing Protocols Configuration Guide, and the Junos OS Feature Guides.

This example Layer 2 bridging firewall filter finds any incoming frames with an IEEE 802.1p learned VLAN priority level of either 1 or 2, and then classifies the packet in the **best-effort** default forwarding class.



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

To configure filtering of frames by IEEE 802.1p bits:

1. Configure the firewall filter **filter-learn-vlan-configure-forwarding**:

```
[edit firewall]
family bridge {
  filter filter-learn-vlan-configure-forwarding {
    term 0 {
      from {
        learn-vlan-1p-priority [1 2];
      }
      then forwarding-class best-effort;
    }
  }
}
```

2. Apply the firewall filter **filter-learn-vlan-configure-forwarding** as an input filter to **ge-0/0/0**:

```
[edit interfaces]
ge-0/0/0 {
  unit 0 {
    family bridge {
      filter {
        input filter-learn-vlan-configure-forwarding;
      }
    }
  }
}
```

Related Documentation

- Layer 2 Firewall Filters
- [Firewall Filters for Bridge Domains and VPLS Instances on page 101](#)
- [Example: Configuring Policing and Marking of Traffic Entering a VPLS Core on page 102](#)
- [Example: Configuring Filtering of Frames by MAC Address on page 104](#)
- [Example: Configuring Filtering of Frames by Packet Loss Priority on page 107](#)

Example: Configuring Filtering of Frames by Packet Loss Priority

To configure an MX Series router firewall filter to provide matching on the packet loss priority (PLP) level carried in the frame, use the **loss-priority** or **loss-priority-except** match condition. Packet loss priority matching is available for all protocols. For more detailed information about configuring firewall filters and configuring filter match conditions for Layer 2 bridging traffic on the MX Series routers, see the Routing Policy Configuration Guide.



NOTE: Layer 2 bridging is supported only on the MX Series routers. For more information about how to configure Layer 2 bridging, see the Junos® OS Network Interfaces, the Junos OS Routing Protocols Configuration Guide, and the Junos OS Feature Guides.

This example Layer 2 bridging firewall filter finds any incoming frames with a packet loss priority (PLP) level of **medium-high**, and then classifies the packet in the **expedited-forwarding** default forwarding class.



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

To configure filtering of frames by packet loss priority:

1. Configure the firewall filter **filter-plp-configure-forwarding**:

```
[edit firewall]
family bridge {
  filter filter-plp-configure-forwarding {
    term 0 {
      from {
        loss-priority medium-high;
      }
      then forwarding-class expedited-forwarding;
    }
  }
}
```

2. Configure a Layer 2 bridging domain **bd** for the **ge-0/0/0** interface (that has already been configured at the **[edit interfaces]** hierarchy level):

```
[edit bridge-domains]
bd {
  domain-type bridge {
    interface ge-0/0/0;
  }
}
```

3. Apply the filter **filter-plp-configure-forwarding** as an input filter to the **ge-0/0/0** interface:

```
[edit interfaces]
ge-0/0/0 {
  unit 0 {
    family bridge {
      filter {
        input filter-plp-configure-forwarding;
      }
    }
  }
}
```

**Related
Documentation**

- Layer 2 Firewall Filters
- [Firewall Filters for Bridge Domains and VPLS Instances on page 101](#)
- [Example: Configuring Policing and Marking of Traffic Entering a VPLS Core on page 102](#)
- [Example: Configuring Filtering of Frames by MAC Address on page 104](#)
- [Example: Configuring Filtering of Frames by IEEE 802.1p Bits on page 106](#)

CHAPTER 10

IEEE 802.1ag OAM Connectivity-Fault Management

This chapter discusses the following topics:

- [Ethernet Operations, Administration, and Maintenance on page 109](#)
- [Ethernet OAM Connectivity Fault Management on page 110](#)
- [Example: Configuring Ethernet CFM over VPLS on page 111](#)
- [Example: Configuring Ethernet CFM on Bridge Connections on page 118](#)
- [Example: Configuring Ethernet CFM on Physical Interfaces on page 122](#)

Ethernet Operations, Administration, and Maintenance

This topic provides an overview to help you effectively configure Ethernet Operations, Administration, and Maintenance (OAM) on a network of Juniper Networks MX Series 3D Universal Edge Routers. For more information about configuring OAM parameters on Ethernet interfaces, see the Junos® OS Network Interfaces.

Ethernet OAM provides the tools that network management software and network managers can use to determine how a network of Ethernet links is functioning. Ethernet OAM should:

- Rely only on the media access control (MAC) address or virtual local area network (VLAN) identifier for troubleshooting
- Work independently of the actual Ethernet transport and function over physical Ethernet ports, or a virtual service such as pseudowire, and so on.
- Isolate faults over a flat (or single operator) network architecture or a nested or hierarchical (or multi-provider) networks.

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.

Ethernet OAM functions are implemented as:

- Fault detection and notification (provided by continuity check messages)
- Path discovery (provided by the linktrace protocol)

- Fault isolation, verification, and recovery (isolation and verification are provided by a combination of protocols, while recovery is the function of protocols such as spanning tree)

The loopback protocol used in Ethernet OAM is modeled on the standard IP ping. After a fault is detected, the loopback protocol performs fault verification and isolation under the direction of a network operator. The loopback is performed using request and response message pairs. A unicast loopback message is generated by a MEP and a loopback reply is generated by the destination MIP or MEP. The target MAC address is learned by the continuity check protocol or linktrace protocol. The loopback message's packet is always forwarded to a unique port by the originating MEP, as determined by a MAC table lookup or the MEP interface MAC address. The target MIP or MEP generates a unicast loopback reply in response to the received loopback message. The loopback message follows the same path as a data packet, and intermediate bridges simply forward the packet to the destination MIP or MEP.

**Related
Documentation**

- Ethernet OAM
- [Ethernet OAM Connectivity Fault Management on page 110](#)
- [Example: Configuring Ethernet CFM over VPLS on page 111](#)
- [Example: Configuring Ethernet CFM on Bridge Connections on page 118](#)
- [Example: Configuring Ethernet CFM on Physical Interfaces on page 122](#)

Ethernet OAM Connectivity Fault Management

The most complete connectivity fault management (CFM) is defined in IEEE 802.1ag. This topic emphasizes the use of CFM in a Metro Ethernet environment.

The major features of CFM are:

- Fault monitoring using the continuity check protocol. This is a neighbor discovery and health check protocol which discovers and maintains adjacencies at the VLAN or link level.
- Path discovery and fault verification using the linktrace protocol. Similar to IP traceroute, this protocol maps the path taken to a destination MAC address through one or more bridged networks between the source and destination.
- Fault isolation using the loopback protocol. Similar to IP ping, this protocol works with the continuity check protocol during troubleshooting.

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 maintenance domain level from 0 through 7. Level allocation is based on the network hierarchy, where outermost domains are assigned a higher level than the innermost domains. Customer end points have to highest maintenance domain level. 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. There is also a maintenance intermediate point (MIP), which is a CFM entity similar to the MEP, but more passive (MIPs only respond to CFM messages).

MEPs can be *up MEPs* or *down MEPs*. A link can connect a MEP at level 5 to a MEP at level 7. The interface at level 5 is an up MEP (because the other end of the link is at MEP level 7) and the interface at level 7 is a down MEP (because the other end of the link is at MEP level 5).

In a Metro Ethernet network, CFM is commonly 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.

In many Metro Ethernet networks, CFM is used to monitor connectivity over a VPLS and bridge network.

Related Documentation

- Ethernet OAM
- [Ethernet Operations, Administration, and Maintenance on page 109](#)
- [Example: Configuring Ethernet CFM over VPLS on page 111](#)
- [Example: Configuring Ethernet CFM on Bridge Connections on page 118](#)
- [Example: Configuring Ethernet CFM on Physical Interfaces on page 122](#)

Example: Configuring Ethernet CFM over VPLS

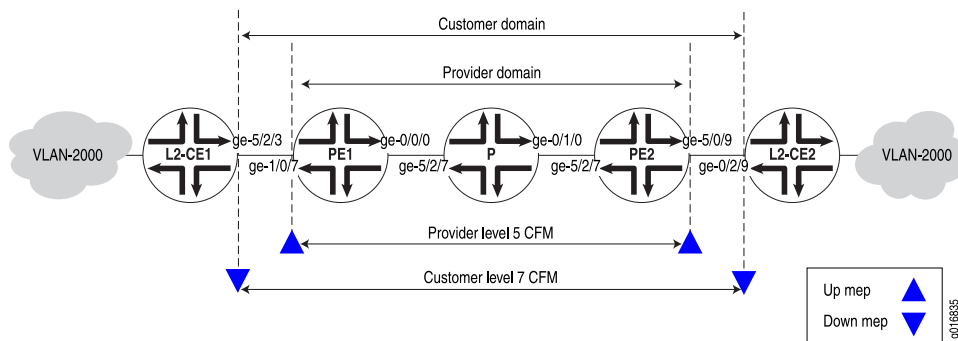
In this example, both the customer and service provider are running Ethernet CFM over a VPLS and a multiprotocol label switching (MPLS) network. The network is shown in [Figure 15 on page 112](#). The customer has configured Ethernet CFM on MX Series routers L2-CE1 and L2-CE2. The service provider has configured Ethernet CFM on MX Series routers PE1, P, and PE2.



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

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 15: 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 {
      interface ge-0/2/9.0;
      direction down;
      auto-discovery;
    }
  }
}
}
}

```

- Related Documentation**
- Ethernet OAM
 - [Ethernet Operations, Administration, and Maintenance on page 109](#)
 - [Ethernet OAM Connectivity Fault Management on page 110](#)
 - [Example: Configuring Ethernet CFM on Bridge Connections on page 118](#)
 - [Example: Configuring Ethernet CFM on Physical Interfaces on page 122](#)

Example: Configuring Ethernet CFM on Bridge Connections

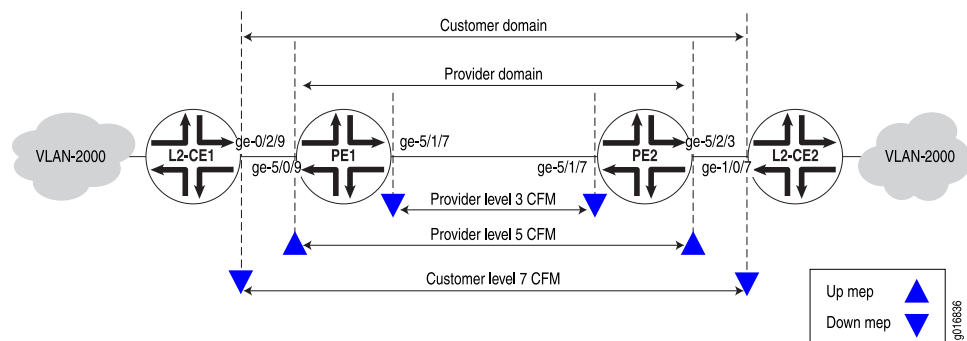
In this example, both the customer and service provider are running Ethernet CFM over a simple bridge network. The network is shown in [Figure 16 on page 118](#). The customer has configured Ethernet CFM on MX Series routers L2-CE1 and L2-CE2. The service provider has configured Ethernet CFM on MX Series routers PE1 and PE2.



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

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 16: 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 protocols oam ethernet]
connectivity-fault-management {
  maintenance-domain customer {
    level 7;
    maintenance-association customer-site1 {
      continuity-check {
        interval 1s;
      }
      mep 700 {
        interface 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 protocols 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;
        }
    }
}

```

- Related Documentation**
- Ethernet OAM
 - [Ethernet Operations, Administration, and Maintenance on page 109](#)
 - [Ethernet OAM Connectivity Fault Management on page 110](#)

- [Example: Configuring Ethernet CFM over VPLS on page 111](#)
- [Example: Configuring Ethernet CFM on Physical Interfaces on page 122](#)

Example: Configuring 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 the IEEE 802.3ah LFM protocol.

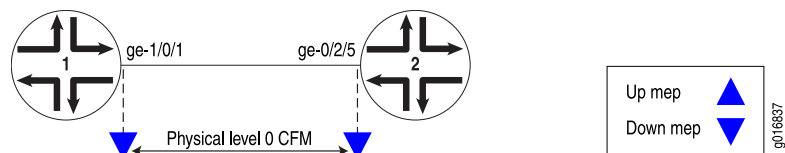
In Junos OS Release 9.3 and later, CFM also supports aggregated Ethernet interfaces. On interfaces configured on Modular Port Concentrators (MPCs) and Modular Interface Cards (MICs) on MX Series routers, CFM is not supported on untagged aggregated Ethernet member links. MPCs and MICs do support CFM on untagged and tagged aggregated Ethernet logical interfaces.



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

In the following example, 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 17 on page 122](#). The single boundary is a “down mep” in CFM terminology.

Figure 17: 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, with the exception of the *mep-id*.

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 200 {
                            interface ge-0/2/5;
                            direction down;
                            auto-discovery;
                        }
                    }
                }
            }
        }
    }
}

```

- Related Documentation**
- Ethernet OAM
 - [Ethernet Operations, Administration, and Maintenance on page 109](#)
 - [Ethernet OAM Connectivity Fault Management on page 110](#)
 - [Example: Configuring Ethernet CFM over VPLS on page 111](#)
 - [Example: Configuring Ethernet CFM on Bridge Connections on page 118](#)

ITU-T Y.1731 Ethernet Frame Delay Measurements

This chapter discusses the following topics:

- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support Ethernet Frame Delay Measurements on page 128](#)
- [Triggering an Ethernet Frame Delay Measurements Session on page 129](#)
- [Viewing Ethernet Frame Delay Measurements Statistics on page 130](#)
- [Example: Configuring One-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces on page 131](#)
- [Example: Configuring Two-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces on page 136](#)
- [Example: Configuring Ethernet Frame Delay Measurements with Untagged Interfaces on page 140](#)

Ethernet Frame Delay Measurements

Performance management depends on the accurate measurement of service agreement objective parameters, which can include bandwidth and reliability. In many cases, a service provider could be subject to penalties imposed by regulation, statute, or contract if network performance is not within the bounds established for the service. One key performance objective is delay, along with its close relative, delay variation (often called jitter). Some applications will function just as well with high delays across the network and high delay variations (such as bulk file transfer), while other applications (such as voice) can only function with low and stable delays. Many networks invoke protocols or features available at Layer 3 (the packet layer) or higher to measure network delays and jitter link-by-link. However, when the network consists of many Ethernet links, there is little available at Layer 2 (the frame layer) that allows routers to measure frame delay and jitter. This is where the ability to configure and monitor Ethernet frame delay is helpful.

On a Juniper Networks MX Series Ethernet Services Router equipped with the Distributed Port Concentrator (MX-DPC) only, you can perform Ethernet frame delay measurements (referred to as ETH-DM in Ethernet specifications). This feature allows you to configure on-demand Operation, Administration, and Maintenance (OAM) statements for the measurement of frame delay and frame delay variation (jitter). You can configure Ethernet

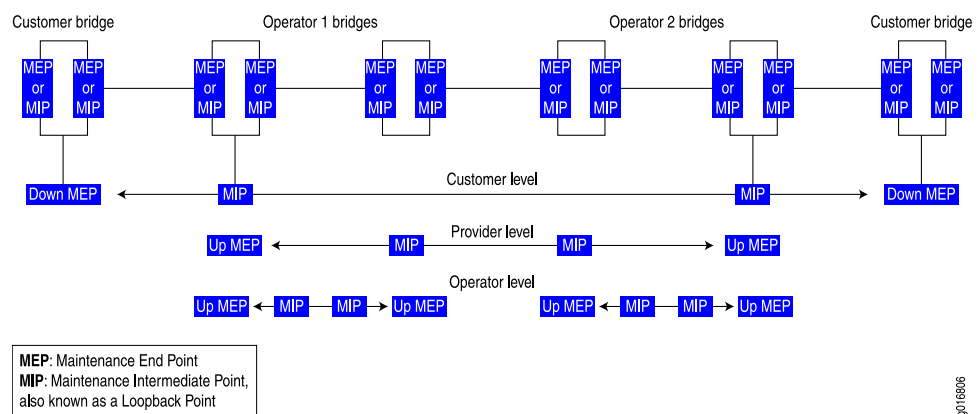
frame delay measurement in either one-way or two-way (round-trip) mode to gather frame delay statistics and simultaneous statistics from multiple sessions. Ethernet frame delay measurement provides fine control to operators for triggering delay measurement on a given service and can be used to monitor Service Level Agreements (SLAs).

Ethernet frame delay measurement also collects other useful information, such as worst and best case delays, average delay, and average delay variation. Ethernet frame delay measurement supports hardware-based timestamping in the receive direction for delay measurements. It also provides runtime display of delay statistics when two-way delay measurement is triggered. Ethernet frame delay measurement records the last 100 samples collected per remote maintenance end point (MEP) or per connectivity fault management (CFM) session. You can retrieve the history at any time using simple commands. You can clear all Ethernet frame delay measurement statistics and PDU counters. Ethernet frame delay measurement is fully compliant with the ITU-T Y.1731 (*OAM Functions and Mechanisms for Ethernet-based Networks*) specification.

Ethernet frame delay measurement uses the IEEE 802.1ag CFM infrastructure.

An overview of the architecture established for Ethernet OAM is shown in [Figure 18 on page 126](#). Generally, Ethernet frame delay measurements are made in a peer fashion from one MEP or CFM session to another. However, these measurements are not made to Maintenance Intermediate Points (MIPs).

Figure 18: Ethernet OAM Overview



There are two types of Ethernet frame delay measurements:

- One-way
- Two-way (round-trip)

For one-way Ethernet frame delay measurement, either MEP can send a request to begin a one-way delay measurement to its peer MEP. However, the statistics are collected only at the receiver MEP. This feature requires the clocks at the transmitting and receiving MEPs to be synchronized. If these clocks fall out of synchronization, only one-way delay variation and average delay variation values are computed correctly (and therefore valid). Use the show commands at the receiver MEP to display one-way delay statistics.

For two-way (round-trip) Ethernet frame delay measurement, either MEP can send a request to begin a two-way delay measurement to its peer MEP, which responds with timestamp information. Run-time statistics are collected and displayed at the initiator MEP. The clocks do not need to be synchronized at the transmitting and receiving MEPs. The Junos OS supports the optional timestamps in delay measurement reply (DMR) frames to increase the accuracy of delay calculations. The Junos OS also supports hardware-assisted timestamping for Ethernet frame delay protocol data units (PDUs) in the reception path.

Use the **show** commands at the initiator MEP to display two-way delay statistics, and at the receiver MEP to display one-way delay statistics.

The following are some limitations with regard to using Ethernet frame delay measurement:

- This feature is available only on MX Series routers.
- Ethernet frame delay measurements are available only when the distributed periodic packet management daemon (ppmd) is enabled.
- The statistics collected are lost after graceful Routing Engine switchover (GRES).
- You can monitor only one session to the same remote MEP or MAC address.
- Accuracy is compromised when the system changes (such as from reconfiguration). We recommend performing Ethernet frame delay measurements on a stable system.
- The use of Ethernet frame delay measurements on aggregated Ethernet and pseudowire interfaces is not supported.
- The use of hardware-assisted timestamping is not supported on all MX DPCs (Rev-B or higher is required).
- If you attempt to perform Ethernet frame delay measurements to a non-MX partner, the incoming Ethernet frame delay PDUs are discarded silently. Ethernet delay measurement commands and capabilities are not available on non-MX Series routers.

**Related
Documentation**

- Ethernet OAM
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Configuring MEP Interfaces to Support Ethernet Frame Delay Measurements

Ethernet frame delay measurement is a useful tool for providing performance statistics or supporting or challenging Service Level Agreements (SLAs). By default, Ethernet frame delay measurement uses software for timestamping and delay calculations. You can optionally use hardware timing to assist in this process and increase the accuracy of the delay measurement results. This assistance is available on the reception path.

Before you can perform Ethernet frame delay measurements on MX Series routers, you must have done the following:

- Configured Ethernet OAM and CFM correctly
- Prepared the measurement between two compatibly configured MX Series routers
- Enabled the distributed periodic packet management daemon (ppmd)
- Avoided trying to perform Ethernet frame delay measurement on aggregated Ethernet or pseudowire interfaces, which are not supported
- Made sure the hardware-assisted timestamping is supported if that feature is configured

At the end of this configuration, you create two MX Series routers that can perform and display Ethernet frame delay measurements on Ethernet interfaces using optional hardware timestamping. By default, Ethernet frame delay measurement uses software for timestamping and delay calculations. You can optionally use hardware timing to assist in this process and increase the accuracy of the delay measurement results. This assistance is available on the reception path.

To configure hardware-assisted timestamping:

1. To enable Ethernet frame delay measurement hardware assistance on the reception path, include the **hardware-assisted-timestamping** statement at the **[edit protocols oam ethernet connectivity-fault-management performance-monitoring]** hierarchy level:

```
[edit]
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        performance-monitoring {
          hardware-assisted-timestamping; # Enable timestamping in hardware.
        }
      }
    }
  }
}
```

2. Ethernet frame delay measurement requires that distributed PPMD is enabled. Before you can gather statistics for Ethernet frame delay measurement, you must make sure that PPMD is configured properly. Without distributed PPMD, delay measurement results are not valid.

To perform Ethernet frame delay measurement, make sure that the following configuration statement is *NOT* present:

```
[edit routing-options]
ppm {
  no-delegate-processing; # This turns distributed PPMD OFF.
}
```

Related Documentation

- Ethernet OAM
- [Ethernet Frame Delay Measurements on page 125](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Triggering an Ethernet Frame Delay Measurements Session

Before Ethernet frame delay measurement statistics can be displayed, they must be collected. To trigger Ethernet frame delay measurement, use the **monitor ethernet delay-measurement (one-way | two-way) (remote-mac-address | mep identifier) maintenance-domain name maintenance-association ma-id [count count] [wait time]** operational command.

The fields for this command are described in [Table 3 on page 129](#).

Table 3: Monitor Ethernet Delay Command Parameters

Parameter	Parameter Range	Description
one-way or two-way	NA	Perform a one-way or two-way (round-trip) delay measurement.
remote-mac-address	Unicast MAC address	Send delay measurement frames to the destination unicast MAC address (use the format xx:xx:xx:xx:xx:xx). Multicast MAC addresses are not supported.
mep identifier	1–8191	The MEP identifier to use for the measurement. The discovered MAC address for this MEP identifier is used.
maintenance-domain name	Existing MD name	Specifies an existing maintenance domain (MD) to use for the measurement.
maintenance-association ma-id	Existing MA identifier	Specifies an existing maintenance association (MA) identifier to use for the measurement.
count count	1–65535 (default: 10)	(Optional) Specifies the number of Ethernet frame delay frames to send. The default is 10.

Table 3: Monitor Ethernet Delay Command Parameters (*continued*)

Parameter	Parameter Range	Description
wait <i>time</i>	1–255 seconds (default: 1)	(Optional) Specifies the number of seconds to wait between frames. The default is 1 second.

If you attempt to monitor delays to a nonexistent MAC address, you must exit the application manually using ^C:

```
user@host> monitor ethernet delay-measurement two-way 00:11:22:33:44:55
Two-way ETH-DM request to 00:11:22:33:44:55, Interface ge-5/2/9.0
^C
--- Delay measurement statistics ---
Packets transmitted: 10, Valid packets received: 0
Average delay: 0 usec, Average delay variation: 0 usec
Best case delay: 0 usec, Worst case delay: 0 usec
```

Related Documentation

- Ethernet OAM
- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Viewing Ethernet Frame Delay Measurements Statistics

Once Ethernet frame delay measurement statistics have been collected, they can be displayed.

To retrieve the last 100 Ethernet frame delay measurement statistics per remote MEP or per CFM session, two types of **show** commands are provided:

- For all OAM frame counters and Ethernet frame delay measurement statistics
- For Ethernet frame delay measurement statistics only

To retrieve all Ethernet frame delay measurement statistics for a given session, use the **show oam ethernet connectivity-fault-management mep-statistics maintenance-domain *name* maintenance-association *name* [local-mep *identifier*] [remote-mep *identifier*] [count *count*]** command.

To retrieve only Ethernet frame delay measurement statistics for a given session, use the **show oam ethernet connectivity-fault-management delay-statistics maintenance-domain *name* maintenance-association *name* [local-mep *identifier*] [remote-mep *identifier*] [count *count*]** command.



NOTE: The only difference in the two commands is the use of the **mep-statistics** and **delay-statistics** keyword.

The fields for these commands are described in [Table 4 on page 131](#).

Table 4: Show Ethernet Delay Command Parameters

Parameter	Parameter Range	Description
maintenance-domain <i>name</i>	Existing MD name	Specifies an existing maintenance domain (MD) to use.
maintenance-association <i>ma-id</i>	Existing MA identifier	Specifies an existing maintenance association (MA) identifier to use.
local-mep <i>identifier</i>	1–8191	When a MEP has been specified, display statistics only for the local MEP.
remote-mep <i>identifier</i>	1–8191	When a MEP has been specified, display statistics only for the discovered MEP.
count <i>count</i>	1–100 (default:100)	The number of entries to display in the results table. By default, all 100 entries are displayed if they exist.



NOTE: For each MEP, you will see frame counters for sent and received Ethernet frame delay measurement frames whenever MEP statistics are displayed.

Related Documentation

- [Ethernet OAM](#)
- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Example: Configuring One-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces

This example uses two MX Series routers: **MX-1** and **MX-2**. The configuration creates a CFM down MEP session on a VLAN-tagged logical interface connecting the two (**ge-5/2/9** on Router **MX-1** and **ge-0/2/5** on Router **MX-2**).



NOTE: These are not complete router configurations.

Configuration on Router **MX-1**:

```
[edit]
interfaces {
  ge-5/2/9 {
    vlan-tagging;
    unit 0 {
      vlan-id 512;
    }
  }
}
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        traceoptions {
          file eoam_cfm.log size 1g files 2 world-readable;
          flag all;
        }
        linktrace {
          path-database-size 255;
          age 10s;
        }
        maintenance-domain md6 {
          level 6;
          maintenance-association ma6 {
            continuity-check {
              interval 100ms;
              hold-interval 1;
            }
            mep 201 {
              interface ge-5/2/9.0;
              direction down;
              auto-discovery;
            }
          }
        }
      }
    }
  }
}
```

Configuration on Router **MX-2**:

```
[edit]
interfaces {
  ge-0/2/5 {
    vlan-tagging;
    unit 0 {
      vlan-id 512;
    }
  }
}
```



```

}
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        traceoptions {
          file eoam_cfm.log size 1g files 2 world-readable;
          flag all;
        }
        linktrace {
          path-database-size 255;
          age 10s;
        }
        maintenance-domain md6 {
          level 6;
          maintenance-association ma6 {
            continuity-check {
              interval 100ms;
              hold-interval 1;
            }
            mep 101 {
              interface ge-0/2/5.0;
              direction down;
              auto-discovery;
            }
          }
        }
      }
    }
  }
}

```

From Router **MX-2**, start a one-way delay measurement to Router **MX-1**.

```

user@MX-2> monitor ethernet delay-measurement one-way mep 201 maintenance-domain md6
maintenance-association ma6 count 10
One-way ETH-DM request to 00:90:69:0a:43:94, Interface ge-0/2/5.0
1DM Frames sent : 10
--- Delay measurement statistics ---
Packets transmitted: 10
Average delay: NA, Average delay variation: NA
Best case delay: NA, Worst case delay: NA

```

The counters are displayed as part of the local MEP database on Router **MX-2**.

```

user@MX-2> show oam ethernet connectivity-fault-management mep-database
maintenance-domain md6 maintenance-domain ma6
Maintenance domain name: md6, Format: string, Level: 6
Maintenance association name: ma6, Format: string
Continuity-check status: enabled, Interval: 100ms, Loss-threshold: 3 frames
MEP identifier: 101, Direction: down, MAC address: 00:90:69:0a:48:57
Auto-discovery: enabled, Priority: 0
Interface name: ge-0/2/5.0, Interface status: Active, Link status: Up
Defects:
  Remote MEP not receiving CCM                : no
  Erroneous CCM received                       : no
  Cross-connect CCM received                   : no
  RDI sent by some MEP                         : no

```

```

Statistics:
  CCMS sent                      : 1590
  CCMS received out of sequence  : 0
  LBMS sent                      : 0
  Valid in-order LBRs received   : 0
  Valid out-of-order LBRs received : 0
  LBRs received with corrupted data : 0
  LBRs sent                     : 0
  LTMS sent                     : 0
  LTMS received                 : 0
  LTRs sent                    : 0
  LTRs received                 : 0
  Sequence number of next LTM request : 0
  1DMs sent                    : 10
  Valid 1DMs received           : 0
  Invalid 1DMs received          : 0
  DMMs sent                    : 0
  DMRs sent                    : 0
  Valid DMRs received           : 0
  Invalid DMRs received          : 0
Remote MEP count: 1
  Identifier  MAC address      State  Interface
    201      00:90:69:0a:43:94   ok    ge-0/2/5.0

```

The remote MEP database statistics are available on Router **MX-1**.

```

user@MX-1> show oam ethernet connectivity-fault-management mep-database
maintenance-domain md6
Maintenance domain name: md6, Format: string, Level: 6
Maintenance association name: ma6, Format: string
Continuity-check status: enabled, Interval: 100ms, Loss-threshold: 3 frames
MEP identifier: 201, Direction: down, MAC address: 00:90:69:0a:43:94
Auto-discovery: enabled, Priority: 0
Interface name: ge-5/2/9.0, Interface status: Active, Link status: Up
Defects:
  Remote MEP not receiving CCM      : no
  Erroneous CCM received            : no
  Cross-connect CCM received        : no
  RDI sent by some MEP              : no
Statistics:
  CCMS sent                      : 1572
  CCMS received out of sequence    : 0
  LBMS sent                      : 0
  Valid in-order LBRs received     : 0
  Valid out-of-order LBRs received : 0
  LBRs received with corrupted data : 0
  LBRs sent                     : 0
  LTMS sent                     : 0
  LTMS received                 : 0
  LTRs sent                    : 0
  LTRs received                 : 0
  Sequence number of next LTM request : 0
  1DMs sent                    : 0
  Valid 1DMs received           : 10
  Invalid 1DMs received          : 0
  DMMs sent                    : 0
  DMRs sent                    : 0
  Valid DMRs received           : 0
  Invalid DMRs received          : 0
Remote MEP count: 1

```

Identifier	MAC address	State	Interface
101	00:90:69:0a:48:57	ok	ge-5/2/9.0

The remote Router **MX-1** should also collect the delay statistics (up to 100 per session) for display with **mep-statistics** or **delay-statistics**.

```
user@MX-1> show oam ethernet connectivity-fault-management mep-statistics
maintenance-domain md6
```

```
MEP identifier: 201, MAC address: 00:90:69:0a:43:94
```

```
Remote MEP count: 1
```

CCMs sent	: 3240
CCMs received out of sequence	: 0
LBMs sent	: 0
Valid in-order LBRs received	: 0
Valid out-of-order LBRs received	: 0
LBRs received with corrupted data	: 0
LBRs sent	: 0
LTMs sent	: 0
LTMs received	: 0
LTRs sent	: 0
LTRs received	: 0
Sequence number of next LTM request	: 0
1DMs sent	: 0
Valid 1DMs received	: 10
Invalid 1DMs received	: 0
DMMs sent	: 0
DMRs sent	: 0
Valid DMRs received	: 0
Invalid DMRs received	: 0

```
Remote MEP identifier: 101
```

```
Remote MAC address: 00:90:69:0a:48:57
```

```
Delay measurement statistics:
```

Index	One-way delay (usec)	Two-way delay (usec)
1	370	
2	357	
3	344	
4	332	
5	319	
6	306	
7	294	
8	281	
9	269	
10	255	

```
Average one-way delay : 312 usec
```

```
Average one-way delay variation: 11 usec
```

```
Best case one-way delay : 255 usec
```

```
Worst case one-way delay : 370 usec
```

```
user@MX-1> show oam ethernet connectivity-fault-management delay-statistics
maintenance-domain md6
```

```
MEP identifier: 201, MAC address: 00:90:69:0a:43:94
```

```
Remote MEP count: 1
```

```
Remote MAC address: 00:90:69:0a:48:57
```

```
Delay measurement statistics:
```

Index	One-way delay (usec)	Two-way delay (usec)
1	370	

2	357
3	344
4	332
5	319
6	306
7	294
8	281
9	269
10	255

Average one-way delay : 312 usec
Average one-way delay variation: 11 usec
Best case one-way delay : 255 usec



NOTE: When two systems are close to each other, their one-way delay values are very high compared to their two-way delay values. This is because one-way delay measurement requires the timing for the two systems to be synchronized at a very granular level and MX Series routers do not support this granular synchronization. However, two-way delay measurement does not require synchronized timing, making two-way delay measurements more accurate.

**Related
Documentation**

- Ethernet OAM
- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Example: Configuring Two-Way Ethernet Frame Delay Measurements with Single-Tagged Interfaces

This example uses two MX Series routers: **MX-1** and **MX-2**. The configuration creates a CFM down MEP session on a VLAN-tagged logical interface connecting the two (**ge-5/2/9** on Router **MX-1** and **ge-0/2/5** on Router **MX-2**).



NOTE: These are not complete router configurations.

Configuration on Router **MX-1**:

```
[edit]
interfaces {
  ge-5/2/9 {
    vlan-tagging;
    unit 0 {
      vlan-id 512;
```

```

    }
  }
}
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        traceoptions {
          file eoam_cfm.log size 1g files 2 world-readable;
          flag all;
        }
        linktrace {
          path-database-size 255;
          age 10s;
        }
        maintenance-domain md6 {
          level 6;
          maintenance-association ma6 {
            continuity-check {
              interval 100ms;
              hold-interval 1;
            }
            mep 201 {
              interface ge-5/2/9.0;
              direction down;
              auto-discovery;
            }
          }
        }
      }
    }
  }
}

```

Configuration on Router MX-2:

```

[edit]
interfaces {
  ge-0/2/5 {
    vlan-tagging;
    unit 0 {
      vlan-id 512;
    }
  }
}
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        traceoptions {
          file eoam_cfm.log size 1g files 2 world-readable;
          flag all;
        }
        linktrace {
          path-database-size 255;
          age 10s;
        }
      }
    }
  }
}

```

```

    }
    maintenance-domain md6 {
        level 6;
        maintenance-association ma6 {
            continuity-check {
                interval 100ms;
                hold-interval 1;
            }
            mep 101 {
                interface ge-0/2/5.0;
                direction down;
                auto-discovery;
            }
        }
    }
}

```

From Router **MX-1**, start a two-way delay measurement to Router **MX-2**.

```

user@MX-1> monitor ethernet delay-measurement two-way mep 101 maintenance-domain md6
maintenance-association ma6 count 10

```

```

Two-way ETH-DM request to 00:90:69:0a:48:57, Interface ge-5/2/9.0
DMR received from 00:90:69:0a:48:57 Delay: 100 usec Delay variation: 0 usec
DMR received from 00:90:69:0a:48:57 Delay: 92 usec Delay variation: 8 usec
DMR received from 00:90:69:0a:48:57 Delay: 92 usec Delay variation: 0 usec
DMR received from 00:90:69:0a:48:57 Delay: 111 usec Delay variation: 19 usec
DMR received from 00:90:69:0a:48:57 Delay: 110 usec Delay variation: 1 usec
DMR received from 00:90:69:0a:48:57 Delay: 119 usec Delay variation: 9 usec
DMR received from 00:90:69:0a:48:57 Delay: 122 usec Delay variation: 3 usec
DMR received from 00:90:69:0a:48:57 Delay: 92 usec Delay variation: 30 usec
DMR received from 00:90:69:0a:48:57 Delay: 92 usec Delay variation: 0 usec
DMR received from 00:90:69:0a:48:57 Delay: 108 usec Delay variation: 16 usec

```

```

--- Delay measurement statistics ---

```

```

Packets transmitted: 10, Valid packets received: 10
Average delay: 103 usec, Average delay variation: 8 usec
Best case delay: 92 usec, Worst case delay: 122 usec

```

The counters are displayed as part of the MEP database on Router **MX-1** maintenance domain **MD6**.

```

user@MX-1> show oam ethernet connectivity-fault-management mep-database
maintenance-domain md6

```

```

Maintenance domain name: md6, Format: string, Level: 6
Maintenance association name: ma6, Format: string
Continuity-check status: enabled, Interval: 100ms, Loss-threshold: 3 frames
MEP identifier: 201, Direction: down, MAC address: 00:90:69:0a:43:94
Auto-discovery: enabled, Priority: 0
Interface name: ge-5/2/9.0, Interface status: Active, Link status: Up
Defects:
  Remote MEP not receiving CCM                : no
  Erroneous CCM received                       : no
  Cross-connect CCM received                   : no
  RDI sent by some MEP                        : no
Statistics:
  CCMs sent                                   : 894

```

```

CCMs received out of sequence          : 0
LBMs sent                             : 0
Valid in-order LBRs received           : 0
Valid out-of-order LBRs received       : 0
LBRs received with corrupted data      : 0
LBRs sent                             : 0
LTMs sent                             : 0
LTMs received                          : 0
LTRs sent                             : 0
LTRs received                          : 0
Sequence number of next LTM request    : 0
1DMs sent                             : 0
Valid 1DMs received                    : 0
Invalid 1DMs received                  : 0
DMMs sent                             : 10
DMRs sent                             : 0
Valid DMRs received                   : 10
Invalid DMRs received                  : 0
Remote MEP count: 1
Identifier   MAC address   State   Interface
  101      00:90:69:0a:48:57   ok     ge-5/2/9.0

```

The collected MEP statistics are saved (up to 100 per remote MEP or per CFM session) and displayed as part of the MEP statistics on Router **MX-1**.

```

user@MX-1> show oam ethernet connectivity-fault-management mep-statistics
maintenance-domain md6

```

```

MEP identifier: 201, MAC address: 00:90:69:0a:43:94

```

```

Remote MEP count: 1

```

```

CCMs sent                             : 3154
CCMs received out of sequence          : 0
LBMs sent                             : 0
Valid in-order LBRs received           : 0
Valid out-of-order LBRs received       : 0
LBRs received with corrupted data      : 0
LBRs sent                             : 0
LTMs sent                             : 0
LTMs received                          : 0
LTRs sent                             : 0
LTRs received                          : 0
Sequence number of next LTM request    : 0
1DMs sent                             : 0
Valid 1DMs received                    : 0
Invalid 1DMs received                  : 0
DMMs sent                             : 10
DMRs sent                             : 0
Valid DMRs received                   : 10
Invalid DMRs received                  : 0

```

```

Remote MEP identifier: 101

```

```

Remote MAC address: 00:90:69:0a:48:57

```

```

Delay measurement statistics:
Index  One-way delay  Two-way delay
      (usec)         (usec)
  1           100
  2           92
  3           92
  4          111
  5          110
  6          119
  7          122

```

```

      8                      92
      9                      92
     10                     108
Average two-way delay      : 103 usec
Average two-way delay variation: 8 usec
Best case two-way delay    : 92 usec
Worst case two-way delay   : 122 usec

```

The collected delay statistics are also saved (up to 100 per session) and displayed as part of the MEP delay statistics on Router **MX-1**.

```

user@MX-1> show oam ethernet connectivity-fault-management delay-statistics
maintenance-domain md6
MEP identifier: 201, MAC address: 00:90:69:0a:43:94
Remote MEP count: 1

```

```

Remote MAC address: 00:90:69:0a:48:57
Delay measurement statistics:
Index One-way delay Two-way delay
                        (usec)      (usec)
  1                      100
  2                      92
  3                      92
  4                     111
  5                     110
  6                     119
  7                     122
  8                      92
  9                      92
 10                     108
Average two-way delay      : 103 usec
Average two-way delay variation: 8 usec
Best case two-way delay    : 92 usec
Worst case two-way delay   : 122 usec

```

Related Documentation

- [Ethernet OAM](#)
- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring ETH-DM with Untagged Interfaces on page 140](#)

Example: Configuring Ethernet Frame Delay Measurements with Untagged Interfaces

Ethernet frame delay measurements are supported on untagged interfaces. All commands are the same as for tagged interfaces. Only the configurations are different. This section shows the untagged interface configurations for Routers **MX-1** and **MX-2**.



NOTE: These are not complete router configurations.

Untagged interface configuration for Router **MX-1**.

```
[edit]
interfaces {
  ge-5/0/0 {
    unit 0;
  }
  ge-5/2/9 {
    unit 0;
  }
}
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        traceoptions {
          file eoam_cfm.log size 1g files 2 world-readable;
          flag all;
        }
        linktrace {
          path-database-size 255;
          age 10s;
        }
        maintenance-domain md6 {
          level 6;
          maintenance-association ma6 {
            continuity-check {
              interval 100ms;
              hold-interval 1;
            }
            mep 201 {
              interface ge-5/0/0;
              direction down;
              auto-discovery;
            }
          }
        }
      }
    }
  }
}
```

Untagged interface configuration for Router **MX-2**.

```
[edit]
interfaces {
  ge-0/2/2 {
    unit 0;
  }
  ge-0/2/5 {
    unit 0;
  }
}
protocols {
  oam {
    ethernet {
```

```
connectivity-fault-management {
  traceoptions {
    file eoam_cfm.log size 1g files 2 world-readable;
    flag all;
  }
  linktrace {
    path-database-size 255;
    age 10s;
  }
  maintenance-domain md6 {
    level 6;
    maintenance-association ma6 {
      continuity-check {
        interval 100ms;
        hold-interval 1;
      }
      mep 101 {
        interface ge-0/2/2;
        direction down;
        auto-discovery;
      }
    }
  }
}
```

**Related
Documentation**

- Ethernet OAM
- [Ethernet Frame Delay Measurements on page 125](#)
- [Configuring MEP Interfaces to Support ETH-DM on page 128](#)
- [Triggering an ETH-DM Session on page 129](#)
- [Viewing ETH-DM Statistics on page 130](#)
- [Configuring One-Way ETH-DM with Single-Tagged Interfaces on page 131](#)
- [Configuring Two-Way ETH-DM with Single-Tagged Interfaces on page 136](#)

CHAPTER 12

IEEE 802.3ah OAM Link-Fault Management

This chapter discusses the following topics:

- [Ethernet OAM Link Fault Management on page 143](#)
- [Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge on page 144](#)
- [Example: Configuring Ethernet LFM for CCC on page 145](#)
- [Example: Configuring Ethernet LFM for Aggregated Ethernet on page 146](#)
- [Example: Configuring Ethernet LFM with Loopback Support on page 148](#)

Ethernet OAM Link Fault Management

Link Fault Management (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.

Related Documentation

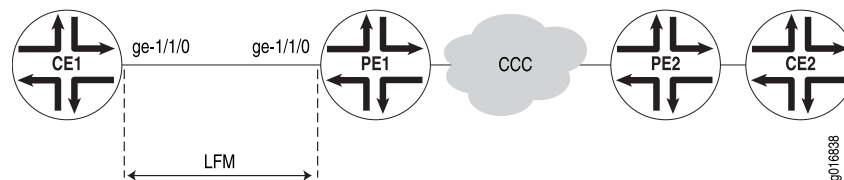
- [Ethernet OAM](#)
- [Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge on page 144](#)
- [Example: Configuring Ethernet LFM for CCC on page 145](#)
- [Example: Configuring Ethernet LFM for Aggregated Ethernet on page 146](#)
- [Example: Configuring Ethernet LFM with Loopback Support on page 148](#)

Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge

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 19 on page 144](#).

Figure 19: Ethernet LFM Between Provider Edge and Customer Edge



To configure Ethernet LFM on an IP link between PE and CE interfaces:

1. Configure LFM on the PE router:

```
[edit]
interfaces ge-1/1/0 {
  unit 0 {
    family inet {
      address 11.11.1.1/24;
    }
  }
}
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface ge-1/1/0 {
          pdu-interval 1000;
          pdu-threshold 5;
        }
      }
    }
  }
}
```

2. Configure LFM on the CE router:

```
[edit]
interfaces ge-1/1/0 {
  unit 0 {
    family inet {
      address 11.11.1.2/24;
    }
  }
}
protocols {
  oam {
    ethernet {
```

```

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

```

Related Documentation

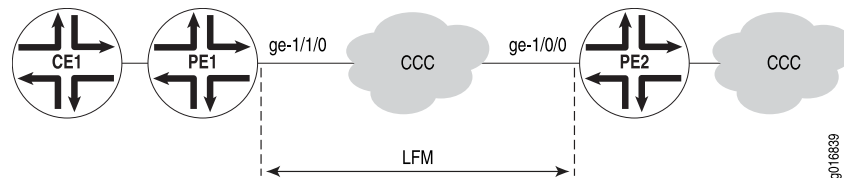
- Ethernet OAM
- [Ethernet OAM Link Fault Management on page 143](#)
- [Example: Configuring Ethernet LFM for CCC on page 145](#)
- [Example: Configuring Ethernet LFM for Aggregated Ethernet on page 146](#)
- [Example: Configuring Ethernet LFM with Loopback Support on page 148](#)

Example: Configuring 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 20 on page 145](#)

Figure 20: Ethernet LFM for CCC



To configure Ethernet LFM between two PEs connected using CCC:

1. 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;
      }
    }
  }
}
```

2. 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;
        }
      }
    }
  }
}
```

Related Documentation

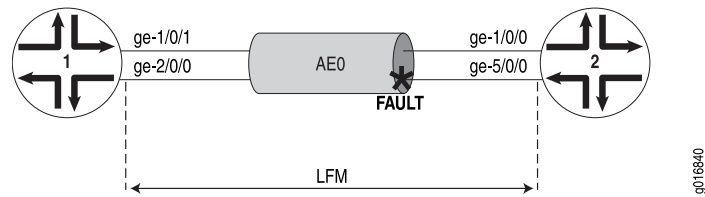
- Ethernet OAM
- [Ethernet OAM Link Fault Management on page 143](#)
- [Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge on page 144](#)
- [Example: Configuring Ethernet LFM for Aggregated Ethernet on page 146](#)
- [Example: Configuring Ethernet LFM with Loopback Support on page 148](#)

Example: Configuring 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 21 on page 147](#).

Figure 21: Ethernet LFM for Aggregated Ethernet



To configure LFM on an aggregated Ethernet interface between two routers:

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;
      }
    }
  }
}
```

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;
      }
    }
  }
}
}

```

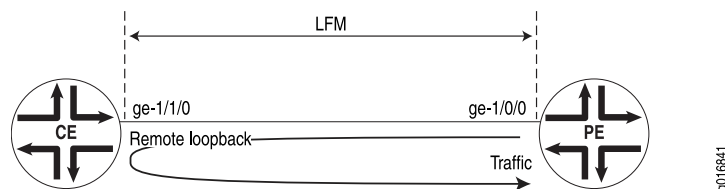
Related Documentation

- Ethernet OAM
- [Ethernet OAM Link Fault Management on page 143](#)
- [Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge on page 144](#)
- [Example: Configuring Ethernet LFM for CCC on page 145](#)
- [Example: Configuring Ethernet LFM with Loopback Support on page 148](#)

Example: Configuring 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 22 on page 148](#).

Figure 22: Ethernet LFM with Loopback Support



To configure LFM between a PE and a CE:

1. 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;
        }
      }
    }
  }
}
```

2. 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;
          }
        }
      }
    }
  }
}
```

- Related Documentation**
- Ethernet OAM
 - [Ethernet OAM Link Fault Management on page 143](#)

- [Example: Configuring Ethernet LFM Between Provider Edge and Customer Edge on page 144](#)
- [Example: Configuring Ethernet LFM for CCC on page 145](#)
- [Example: Configuring Ethernet LFM for Aggregated Ethernet on page 146](#)

CHAPTER 13

Ethernet Ring Protection

This chapter discusses the following topics:

- [Ethernet Ring Protection on page 151](#)
- [Ethernet Ring Protection Using Ring Instances for Load Balancing on page 153](#)
- [Example: Configuring Ethernet Ring Protection for MX Series Routers on page 154](#)
- [Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers on page 160](#)
- [Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation on page 177](#)
- [Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition on page 179](#)

Ethernet Ring Protection

Link failure is often an unavoidable part of networking. However, there are methods of improving the reliability of a router or bridge network even when link failures occur. For example, SONET/SDH seal-healing rings are frequently used to add a level of robustness to router networks. This ring protection switching is now extended to Ethernet links. You can configure Ethernet ring protection for a series of two or more systems so that if one link fails, traffic is rerouted around the failure on the ring.

The basic idea of Ethernet ring protection is to use one specific link to protect the whole ring. This special link is the ring protection link (RPL). When all links are up and running, the RPL blocks traffic and remains idle. The RPL itself is controlled by the designated RPL owner node. There is only one RPL owner node on the ring and the RPL owner node is responsible for blocking the RPL interface under normal operating conditions. However, if a link failure occurs on the ring, the RPL owner node is responsible for unblocking the RPL interface and protection—switching the traffic on the alternate path around the ring. An Ethernet ring automatic protection switching (R-APS) messaging protocol coordinates the protection activities of all nodes on the ring. The APS blocks traffic over the failed link and unblocks traffic over the RPL.

When the failed link is repaired, the traffic reverts to its normal pattern. That is, the RPL owner blocks the RPL link and unblocks traffic over the cleared link.

Two or more nodes form a ring. Links between the nodes form a chain, with the last node also connecting the first. Every ring node therefore has two ports related to the ring, one in each direction. In this chapter, these directions are referred to as east and west.

Every node on the ring is one of two types:

- RPL owner node—This node owns the RPL and blocks or unblocks the RPL as conditions require. This node initiates the R-APS message.
- Normal node—All other nodes on the ring (that is, those that are not the RPL owner node) operate as normal nodes and have no special role on the ring.

In addition to roles, each node on the Ethernet ring can be in one of several states:

- Init—The node is not yet participating in the ring.
- Idle—The node is performing normally (there is no link failure on the ring). In this state, traffic is unblocked on both ring ports, except for the RPL owner node, which blocks the RPL port (the other RPL owner port is unblocked).
- Protection—When a failure occurs on the ring, a normal node will have traffic blocked on the ring port that connects to the failed link. The RPL owner, if it is not at one end of the failed link, will then unblock the RPL port so both ports are active.



NOTE: The R-APS protocol does not detect the number of RPL owner nodes configured on the ring. You must configure only one RPL and RPL owner per ring or protection switching will not work properly.

Ethernet ring protection only works when one link on the ring fails. Multiple link failures will break the ring and cause protection switching to fail.

Several restrictions apply to Ethernet ring protection:

- The Ethernet ring protection configured as a single instance only works at the physical level (adjacent nodes must be directly connected). The ring protection operates at the interface (port) level and not at the VLAN level.
- Manual (command-based) switching to protection mode is not supported.
- Nonrevertive switching is not supported. When the link failure is cleared, traffic always returns to normal operation.
- The interconnection of multiple rings for protection purposes is not supported.

You can configure Ethernet ring protection to optimize traffic load-balancing by using multiple ring instances. For more information about multiple ring instances, see [“Ethernet Ring Protection Using Ring Instances for Load Balancing” on page 153](#)

**Related
Documentation**

- Ethernet OAM
- [Example: Configuring Ethernet Ring Protection for MX Series Routers on page 154](#)
- [Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation on page 177](#)
- [Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition on page 179](#)
- [Ethernet Ring Protection Using Ring Instances for Load Balancing on page 153](#)

- [Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers on page 160](#)

Ethernet Ring Protection Using Ring Instances for Load Balancing

Juniper Network MX Series 3D Universal Edge Routers support Ethernet ring protection (ERP) to help achieve high reliability and network stability. ERP is used in router or bridge networks to protect against link failure. A single-ring topology is configured that uses one specific link called a ring protection link (RPL) to protect the whole ring. When all links are up and running, the RPL blocks traffic and remains idle. However, if a link fails, the RPL routes traffic to bypass the failure on the ring.



NOTE: To learn how ERP works in a single-ring topology, see [“Ethernet Ring Protection” on page 151](#).

MX Series routers now support ERP ring instances. Whereas traffic in a single-ring topology follows the same path, traffic within ring instances allows some traffic to pass through one path while other traffic can follow a different path. Dividing traffic in this way supports traffic load balancing in the physical ring.

Ring instances are like traffic channels that contain different sets of virtual LANS (VLANs). A ring instance is responsible for the protection of a subset of VLANs that transport traffic over the physical ring. When ring instances are configured for the ring, each ring instance should have its own RPL owner, an east and a west interface, and a ring protection link end.

Each ring instance has a control channel and a specific data channel. A data channel is a group of bridge domain VLAN IDs. All VLAN IDs within the same ring interface share the same data-forwarding properties controlled by the ERP. If no data channel is defined in the ring configuration, ERP will only operate on the physical link instead of as a ring instance using logical links.

When operating ERP in a topology with other protocols, the following considerations should be observed:

- If a physical interface is part of an Ethernet ring, it cannot be configured for Spanning Tree Protocol (STP) or Multiple Spanning Tree Protocol (MSTP).
- ERP and Per-VLAN Spanning Tree (PVST) can be configured on the same topology as long as PVST doesn't share the same VLAN with any Ethernet ring instance configured on the physical port.
- If ERP is configured only as a physical ring instance (a ring without a data channel) in a topology also configured for PVST, ERP checks the PVST configuration on two ring interfaces and automatically creates a data channel excluding VLANs used by PVST.

Related Documentation

- [Ethernet Ring Protection on page 151](#)
- [Example: Configuring Ethernet Ring Protection for MX Series Routers on page 154](#)

Example: Configuring Ethernet Ring Protection for MX Series Routers

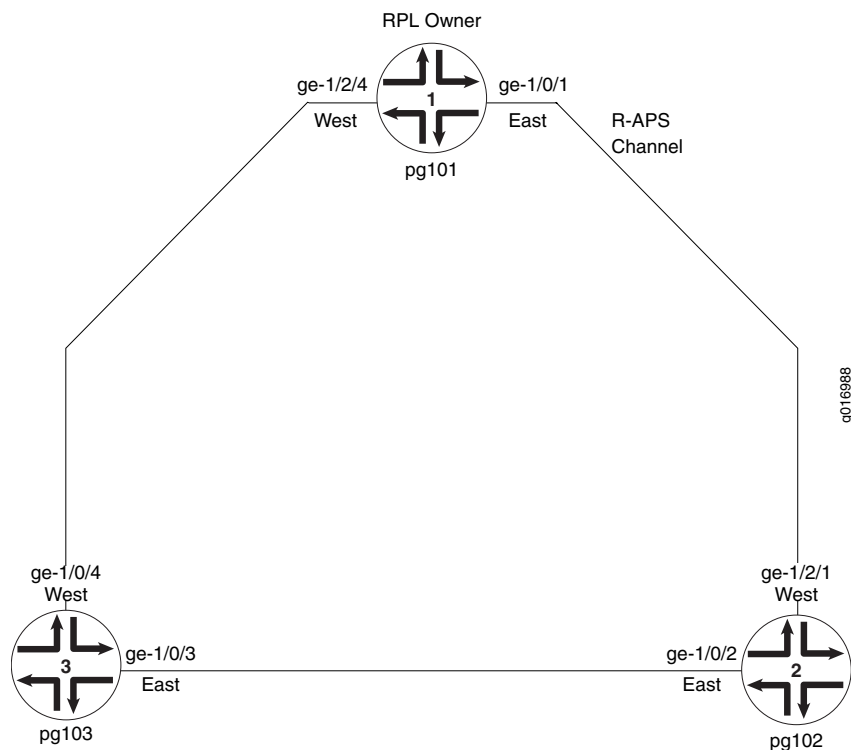
This example configures Ethernet ring protection for three MX Series router nodes:

- [Example Topology on page 154](#)
- [Router 1 \(RPL Owner\) Configuration on page 155](#)
- [Router 2 Configuration on page 157](#)
- [Router 3 Configuration on page 158](#)

Example Topology

The links connecting the three MX Series routers are shown in [Figure 23 on page 154](#).

Figure 23: Ethernet Ring Protection Example Nodes



This example uses the following topology details for Ethernet ring protection:

- Router 1 is the RPL owner. The node identification for Router 1 is MAC address **00:01:01:00:00:01**.
- The RPL link is **ge-1/0/1.1** (this is also the R-APS messaging control channel).
- Traffic flows among the nodes in the configured bridge domains. (That is, only the control channels are configured.)
- Router 1's east control channel interface is **ge-1/0/1.1** (the RPL) and the west control channel interface is **ge-1/2/4.1**. The protection group is **pg101**.

- Router 2's east control channel interface is **ge-1/0/2.1** (the RPL) and the west control channel interface is **ge-1/2/1.1**. The protection group is **pg102**.
- Router 3's east control channel interface is **ge-1/0/3.1** (the RPL) and the west control channel interface is **ge-1/0/4.1**. The protection group is **pg103**.



NOTE: Although not strictly required for physical ring protection, this example configures Ethernet OAM with MEPs.

Router 1 (RPL Owner) Configuration

To configure Router 1 (the RPL owner):

1. Configure the interfaces:

```
[edit]
interfaces {
  ge-1/0/1 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
    }
  }
  ge-1/2/4 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
    }
  }
  irb {
    unit 0 {
      family inet {
        address address 192.1.1.11/24;
      }
    }
  }
}
```

2. Configure the bridge domain:

```
[edit]
bridge-domains {
  bd1 {
    domain-type bridge;
    vlan-id 100;
    interface ge-1/2/4.1;
    interface ge-1/0/1.1;
    routing-interface irb.0;
  }
}
```

3. Configure the Ethernet ring protection group:

```
[edit]
protocols {
  protection-group {
    ethernet-ring pg101 {
      node-id 00:01:01:00:00:01;
      ring-protection-link-owner;
      east-interface {
        control-channel ge-1/0/1.1;
      }
      ring-protection-link-end;
      west-interface {
        control-channel ge-1/2/4.1;
      }
    }
  }
}
```

4. Configure Ethernet OAM:

```
[edit]
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        action-profile rmep-defaults {
          default-action {
            interface-down;
          }
        }
      }
      maintenance-domain d1 {
        level 0;
        maintenance-association 100 {
          mep 1 {
            interface ge-1/0/1;
            remote-mep 2 {
              action-profile rmep-defaults;
            }
          }
        }
      }
      maintenance-domain d2 {
        level 0;
        maintenance-association 100 {
          mep 1 {
            interface ge-1/2/4;
            remote-mep 2 {
              action-profile rmep-defaults;
            }
          }
        }
      }
    }
  }
}
```


Router 2 Configuration

To configure Router 2:

1. Configure the interfaces:

```
[edit]
interfaces {
  ge-1/0/2 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
    }
  }
  ge-1/2/1 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
      encapsulation vlan-bridge;
      vlan-id 100;
    }
  }
  irb {
    unit 0 {
      family inet {
        address address 192.1.1.22/24;
      }
    }
  }
}
```

2. Configure the bridge domain:

```
[edit]
bridge-domains {
  bd1 {
    domain-type bridge;
    vlan-id 100;
    interface ge-1/2/1.1;
    interface ge-1/0/2.1;
    routing-interface irb.0;
  }
}
```

3. Configure the Ethernet protection group:

```
[edit]
protocols {
  protection-group {
    ethernet-ring pg102 {
      node-id 00:22:22:22:22:22;
      east-interface {
        control-channel ge-1/0/2.1;
      }
      west-interface {
```

```
        control-channel ge-1/2/1.1;
      }
    }
  }
}
```

4. Configure Ethernet OAM:

```
[edit]
protocols {
  oam {
    ethernet {
      connectivity-fault-management {
        action-profile rmep-defaults {
          default-action {
            interface-down;
          }
        }
      }
      maintenance-domain d1 {
        level 0;
        maintenance-association 100 {
          mep 2 {
            interface ge-1/2/1;
            remote-mep 1 {
              action-profile rmep-defaults;
            }
          }
        }
      }
      maintenance-domain d3 {
        level 0;
        maintenance-association 100 {
          mep 1 {
            interface ge-1/0/2;
            remote-mep 2 {
              action-profile rmep-defaults;
            }
          }
        }
      }
    }
  }
}
```

Router 3 Configuration

To configure Router 3:

1. Configure the interfaces:

```
[edit]
interfaces {
  ge-1/0/4 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
```

```

    unit 1 {
        encapsulation vlan-bridge;
        vlan-id 100;
    }
}
ge-1/0/3 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
        encapsulation vlan-bridge;
        vlan-id 100;
    }
}
irb {
    unit 0 {
        family inet {
            address 192.1.1.33/24;
        }
    }
}
}

```

2. Configure the bridge domain:

```

[edit]
bridge-domains {
    bd1 {
        domain-type bridge;
        vlan-id 100;
        interface ge-1/0/4.1;
        interface ge-1/0/3.1;
        routing-interface irb.0;
    }
}

```

3. Configure the Ethernet protection group:

```

[edit]
protocols {
    protection-group {
        ethernet-ring pg103 {
            node-id 00:33:33:33:33:33;
            east-interface {
                control-channel ge-1/0/3.1;
            }
            west-interface {
                control-channel ge-1/0/4.1;
            }
        }
    }
}

```

4. Configure Ethernet OAM:

```

[edit]
protocols {
    oam {
        ethernet {

```

```
connectivity-fault-management {
  action-profile rmep-defaults {
    default-action {
      interface-down;
    }
  }
  maintenance-domain d2 {
    level 0;
    maintenance-association 100 {
      mep 2 {
        interface ge-1/0/4;
        remote-mep 1 {
          action-profile rmep-defaults;
        }
      }
    }
  }
  maintenance-domain d3 {
    level 0;
    maintenance-association 100 {
      mep 2 {
        interface ge-1/0/3;
        remote-mep 1 {
          action-profile rmep-defaults;
        }
      }
    }
  }
}
```

Related Documentation

- Ethernet OAM
- [Ethernet Ring Protection on page 151](#)
- [Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation on page 177](#)
- [Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition on page 179](#)

Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers

MX Series routers support Ethernet ring protection (ERP) to help achieve high reliability and network stability. ERP is used in router or bridge networks to protect against link failure. A single-ring topology is configured that uses one specific link called a ring protection link (RPL) to protect the whole ring. When all links are up and running, the RPL blocks traffic and remains idle. However, if a link fails, the RPL routes traffic to bypass the failure on the ring.

MX Series routers now support ERP ring instances. Whereas traffic in a ring topology follows the same path, traffic within a ring instance uses data channels to allow some

traffic to pass through one path while other traffic can follow a different one. Dividing traffic in this way supports traffic load-balancing in the ring.

This example describes how to use ERP with ring instances to load-balance traffic while still providing network protection from link failure:

- [Requirements on page 161](#)
- [Overview and Topology on page 161](#)
- [Configuration on page 164](#)
- [Verification on page 172](#)

Requirements

This example uses the following hardware and software components:

- Two MX Series routers acting as core switches
- One MX Series router acting as an aggregation switch
- Junos OS Release 10.2 or later for MX Series routers

Overview and Topology

[Figure 24 on page 162](#) displays the topology for this example. The topology contains three MX Series routers. CS1 and CS2 act as core routers in the topology, and AS1 acts as an aggregation switch. Each router has two ring instances, ring-1 and ring-2. All nodes on the ring coordinate protection activities by exchanging messages through the Ethernet ring automatic protection switching (R-APS) messaging protocol. Each ring instance has an RPL owner. The ring-1 RPL owner is CS1; the ring-2 RPL owner is CS2. The RPL owners block or unblock the RPL as conditions require and initiate R-APS messages.

Each ring instance has two interface ports (an east interface and a west interface) that participate in the instance. Interface **ge-2/0/8.0**, the west interface on CS2, is the ring protection link end where ring-2's RPL terminates. Interface **ge-3/2/4.0**, the east interface on CS1, is the ring protection link end where ring-1's RPL terminates.

Each ring instance has a data channel. A data channel is a group of bridge domain virtual LAN (VLAN) IDs. All VLAN IDs within the same ring interface share the same data-forwarding properties controlled by the ERP. The data channel on ring-1 is [200, 300]. The data channel on ring-2 is [500, 600].

Two customer site switches are connected to AS1. Customer site 1 uses VLANs 200 and 300. Customer site 2 uses VLANs 500 and 600.

Figure 24: ERP with Multiple Protection Instances Configured on Three MX Series Routers

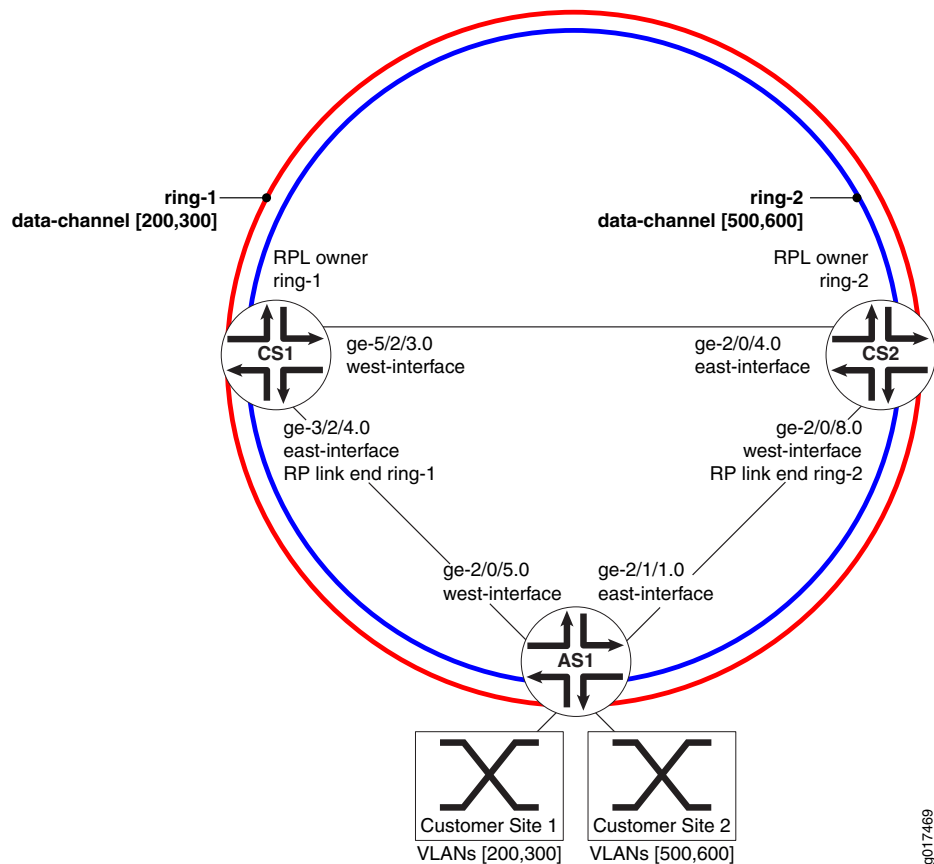


Table 5 on page 162 describes the components of the example topology.

Table 5: Components of the Network Topology

Property	Settings
Ring instances	<ul style="list-style-type: none"> ring-1—Data channel [200,300] ring-2—Data channel [500,600]
Customer sites	<p>Two customer sites are connected to AS 1:</p> <ul style="list-style-type: none"> Customer site 1, VLAN 200 and VLAN 300 Customer site 2, VLAN 500 and VLAN 600

Table 5: Components of the Network Topology (*continued*)

Property	Settings
CS1 router	<p>CS1 has the following protection group properties:</p> <ul style="list-style-type: none"> • RPL owner—ring-1. • East interface—ge-3/2/4.0. • West interface—ge-5/2/3.0. • Data channel for ring-1—VLAN 200, VLAN 300. • Data channel for ring-2—VLAN 500, VLAN 600. • Ring protection link end for ring-1—ge-3/2/4.0. <p>CS1 has the following routing and bridging properties:</p> <ul style="list-style-type: none"> • Routing instance—vs. • Bridge domains: <ul style="list-style-type: none"> • bd100 is associated with vlan-id 100. • bd101 is associated with vlan-id 101. • bd200 is associated with vlan-id 200. • bd300 is associated with vlan-id 300. • bd500 is associated with vlan-id 500. • bd600 is associated with vlan-id 600.
CS2 router	<p>CS2 has the following protection group properties:</p> <ul style="list-style-type: none"> • RPL owner—ring-2. • East interface—ge-2/0/4.0. • West interface—ge-2/0/8.0. • Ring protection link end for ring-2—ge-2/0/8.0. • Data channel for ring-1—VLAN 200, VLAN 300. • Data channel for ring-2—VLAN 500, VLAN 600. <p>CS2 has the following bridging properties:</p> <ul style="list-style-type: none"> • bd100 is associated with vlan-id 100. • bd101 is associated with vlan-id 101. • bd200 is associated with vlan-id 200. • bd300 is associated with vlan-id 300. • bd500 is associated with vlan-id 500. • bd600 is associated with vlan-id 600.

Table 5: Components of the Network Topology (*continued*)

Property	Settings
AS1 router	<p>AS1 has the following protection group properties:</p> <ul style="list-style-type: none"> • East interface—ge-2/0/5.0. • West interface—ge-2/1/1.0. • Data channel for ring-1—VLAN 200, VLAN 300. • Data channel for ring-2—VLAN 500, VLAN 600. <p>AS1 has the following bridging properties:</p> <ul style="list-style-type: none"> • bd100 is associated with vlan-id 100. • bd101 is associated with vlan-id 101. • bd200 is associated with vlan-id 200. • bd300 is associated with vlan-id 300. • bd500 is associated with vlan-id 500. • bd600 is associated with vlan-id 600.

Configuration

To enable ERP with ring instances on CS1, CS2, and AS1, perform these tasks:

- [Configuring ERP on CS1 on page 164](#)
- [Configuring ERP on CS2 on page 167](#)
- [Configuring ERP on AS1 on page 170](#)

Configuring ERP on CS1

CLI Quick Configuration

To quickly configure CS1 for ERP, copy the following commands and paste them into the switch terminal window of CS1:

```
[edit]
set interfaces ge-3/2/4 vlan-tagging
set interfaces ge-3/2/4 unit 0 family bridge interface-mode trunk
set interfaces ge-3/2/4 unit 0 family bridge vlan-id-list 100-1000
set interfaces ge-5/2/3 vlan-tagging
set interfaces ge-5/2/3 unit 0 family bridge interface-mode trunk
set interfaces ge-5/2/3 unit 0 family bridge vlan-id-list 100-1000
set protocols protection-group ethernet-ring ring-1 ring-protection-link-owner
set protocols protection-group ethernet-ring ring-1 east-interface control-channel ge-3/2/4.0
set protocols protection-group ethernet-ring ring-1 east-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 east-interface ring-protection-link-end
set protocols protection-group ethernet-ring ring-1 west-interface control-channel ge-5/2/3.0
set protocols protection-group ethernet-ring ring-1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 data-channel vlan [200, 300]
set protocols protection-group ethernet-ring ring-2 east-interface control-channel ge-3/2/4.0
set protocols protection-group ethernet-ring ring-2 east-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 west-interface control-channel ge-5/2/3.0
set protocols protection-group ethernet-ring ring-2 west-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 data-channel vlan [500, 600]
set routing-instances vs instance-type virtual-switch
set routing-instances vs interface ge-3/2/4.0
set routing-instances vs interface ge-5/2/3.0
set routing-instances vs bridge-domains bd101 vlan-id 101
```



```

set routing-instances vs bridge-domains bd200 vlan-id 200
set routing-instances vs bridge-domains bd300 vlan-id 300
set routing-instances vs bridge-domains bd500 vlan-id 500
set routing-instances vs bridge-domains bd600 vlan-id 600

```

Step-by-Step Procedure To configure ERP on CS1:

1. Configure the trunk interface **ge-3/2/4** to connect CS1 to CS2 and the trunk interface **ge-5/2/3** to connect CS1 to AS, and configure the **family** statement as **bridge** with a VLAN ID list of 100 through 1000:

```

[edit interfaces]
user@cs1# set ge-3/2/4 vlan-tagging
user@cs1# set ge-3/2/4 unit 0 family bridge interface-mode trunk
user@cs1# set ge-3/2/4 unit 0 family bridge vlan-id-list 100-1000
user@cs1# set ge-5/2/3 vlan-tagging
user@cs1# set ge-5/2/3 unit 0 family bridge interface-mode trunk
user@cs1# set ge-5/2/3 unit 0 family bridge vlan-id-list 100-1000

```

2. Enable ERP, specifying the control channels and data channels for **ring-1** and **ring-2**, and configure **ring-1** as the ring protection link owner:



NOTE: Always configure the east-interface statement first, before configuring the west-interface statement.

```

[edit protection-group]
user@cs1# set ethernet-ring ring-1 ring-protection-link-owner
user@cs1# set ethernet-ring ring-1 east-interface control-channel ge-3/2/4.0
user@cs1# set ethernet-ring ring-1 east-interface control-channel vlan 100
user@cs1# set ethernet-ring ring-1 east-interface ring-protection-link-end
user@cs1# set ethernet-ring ring-1 west-interface control-channel ge-5/2/3.0
user@cs1# set ethernet-ring ring-1 west-interface control-channel vlan 100
user@cs1# set ethernet-ring ring-1 data-channel vlan [200, 300]
user@cs1# set ethernet-ring ring-2 east-interface control-channel ge-3/2/4.0
user@cs1# set ethernet-ring ring-2 east-interface control-channel vlan 101
user@cs1# set ethernet-ring ring-2 west-interface control-channel ge-5/2/3.0
user@cs1# set ethernet-ring ring-2 west-interface control-channel vlan 101
user@cs1# set ethernet-ring ring-2 data-channel vlan [500, 600]

```

3. Configure the routing instance, the bridge domains, and the VLAN IDs associated with each bridge domain:

```

[edit routing-instances]
user@cs1# set vs instance-type virtual-switch
user@cs1# set vs interface ge-3/2/4.0
user@cs1# set vs interface ge-5/2/3.0
user@cs1# set vs bridge-domains bd100 vlan-id 100
user@cs1# set vs bridge-domains bd101 vlan-id 101
user@cs1# set vs bridge-domains bd200 vlan-id 200
user@cs1# set vs bridge-domains bd300 vlan-id 300
user@cs1# set vs bridge-domains bd500 vlan-id 500
user@cs1# set vs bridge-domains bd600 vlan-id 600

```

Results Check the results of the configuration:

```

user@cs1> show configuration
interfaces {
  ge-3/2/4 {

```

```
vlan-tagging;
unit 0 {
  family bridge {
    interface-mode trunk;
    vlan-id-list 100-1000;
  }
}
ge-5/2/3 {
  vlan-tagging;
  unit 0 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 100-1000;
    }
  }
}
protocols {
  protection-group {
    ethernet-ring ring-1 {
      east-interface {
        control-channel {
          ge-3/2/4.0;
          vlan 100;
        }
      }
      ring-protection-link-end;
    }
    west-interface {
      control-channel {
        ge-5/2/3.0;
        vlan 100;
      }
    }
    data-channel {
      vlan [ 200 300 ];
    }
  }
}
  protection-group {
    ethernet-ring ring-2 {
      east-interface {
        control-channel {
          ge-3/2/4.0;
          vlan 101;
        }
      }
      west-interface {
        control-channel {
          ge-5/2/3.0;
          vlan 101;
        }
      }
      data-channel {
        vlan [ 500 600 ];
      }
    }
  }
}
```

```

    }
  }
  routing-instances {
    vs {
      instance-type virtual-switch;
      interface ge-3/2/4.0;
      interface ge-5/2/3.0;
      bridge-domains {
        bd100 {
          vlan-id 100;
        }
        bd101 {
          vlan-id 101;
        }
        bd200 {
          vlan-id 200;
        }
        bd300 {
          vlan-id 300;
        }
        bd500 {
          vlan-id 500;
        }
        bd600 {
          vlan-id 600;
        }
      }
    }
  }
}

```

Configuring ERP on CS2

CLI Quick Configuration

To quickly configure CS2 for ERP, copy the following commands and paste them into the switch terminal window of CS2:

```

[edit]
set interfaces ge-2/0/4 unit 0 family bridge interface-mode trunk
set interfaces ge-2/0/4 unit 0 family bridge vlan-id-list 100-1000
set interfaces ge-2/0/8 unit 0 family bridge interface-mode trunk
set interfaces ge-2/0/8 unit 0 family bridge vlan-id-list 100-1000
set protocols protection-group ethernet-ring ring-1 east-interface control-channel ge-2/0/4.0
set protocols protection-group ethernet-ring ring-1 east-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 west-interface control-channel ge-2/0/8.0
set protocols protection-group ethernet-ring ring-1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 data-channel vlan [200, 300]
set protocols protection-group ethernet-ring ring-2 ring-protection-link-owner
set protocols protection-group ethernet-ring ring-2 east-interface control-channel ge-2/0/4.0
set protocols protection-group ethernet-ring ring-2 east-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 west-interface control-channel ge-2/0/8.0
set protocols protection-group ethernet-ring ring-2 west-interface ring-protection-link-end
set protocols protection-group ethernet-ring ring-2 west-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 data-channel vlan [500, 600]
set bridge-domains bd100 vlan-id 100
set bridge-domains bd101 vlan-id 101
set bridge-domains bd200 vlan-id 200
set bridge-domains bd300 vlan-id 300

```

```
set bridge-domains bd500 vlan-id 500
set bridge-domains bd600 vlan-id 600
```

Step-by-Step Procedure

To configure ERP on CS2:

1. Configure the trunk interface **ge-2/0/4** to connect CS2 to CS1 and trunk interface **ge-2/0/8** to connect CS2 to CS1, and configure the **family** statement as **bridge** with a VLAN ID list of 100 through 1000:

```
[edit interfaces]
user@cs2# set ge-2/0/4 unit 0 family bridge interface-mode trunk
user@cs2# set ge-2/0/4 unit 0 family bridge vlan-id-list 100-1000
user@cs2# set ge-2/0/8 unit 0 family bridge interface-mode trunk
user@cs2# set ge-2/0/8 unit 0 family bridge vlan-id-list 100-1000
```

2. Enable ERP, specifying the control channels and data channels for **ring-1** and **ring-2**, and configure **ring-2** as the ring protection link owner:



NOTE: Always configure the east-interface statement first, before configuring the west-interface statement.

```
[edit protection-group]
user@cs2# set ethernet-ring ring-1 east-interface control-channel ge-2/0/4.0
user@cs2# set ethernet-ring ring-1 east-interface control-channel vlan 100
user@cs2# set ethernet-ring ring-1 west-interface control-channel ge-2/0/8.0
user@cs2# set ethernet-ring ring-1 west-interface control-channel vlan 100
user@cs2# set ethernet-ring ring-2 data-channel vlan [200,300]
user@cs2# set ethernet-ring ring-2 east-interface control-channel ge-2/0/4.0
user@cs2# set ethernet-ring ring-2 east-interface control-channel vlan 101
user@cs2# set ethernet-ring ring-2 ring-protection-link-owner
user@cs2# set ethernet-ring ring-2 west-interface control-channel ge-2/0/8.0
user@cs2# set ethernet-ring ring-2 west-interface control-channel vlan 101
user@cs2# set ethernet-ring ring-2 west-interface ring-protection-link-end
user@cs2# set ethernet-ring ring-2 data-channel vlan [500,600]
```

3. Configure the routing instance, the bridge domains, and the VLAN IDs associated with each bridge domain:

```
[edit bridge-domains]
user@cs2# set bd100 vlan-id 100
user@cs2# set bd101 vlan-id 101
user@cs2# set bd200 vlan-id 200
user@cs2# set bd300 vlan-id 300
user@cs2# set bd500 vlan-id 500
user@cs2# set bd600 vlan-id 600
```

Results Check the results of the configuration:

```
user@cs2> show configuration
interfaces {
  ge-2/0/4 {
    unit 0 {
      family bridge {
        interface-mode trunk;
        vlan-id-list 100-1000;
      }
    }
  }
}
```

```
}
ge-2/0/8 {
  unit 0 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 100-1000;
    }
  }
}
protocols {
  protection-group {
    ethernet-ring ring-1 {
      east-interface {
        control-channel {
          ge-2/0/4.0;
          vlan 100;
        }
      }
      west-interface {
        control-channel {
          ge-2/0/8.0;
          vlan 100;
        }
      }
      data-channel {
        vlan [200, 300];
      }
    }
  }
}
ethernet-ring ring-2 {
  east-interface {
    control-channel {
      ge-2/0/4.0;
      vlan 101;
    }
  }
  west-interface {
    control-channel {
      ge-2/0/8.0;
      vlan 101;
    }
  }
  ring-protection-link-end;
}
data-channel {
  vlan [500, 500];
}
}
bridge-domains {
  bd100 {
    vlan-id 100;
  }
  bd101 {
    vlan-id 101;
  }
  bd200 {
```

```

        vlan-id 200;
    }
    bd300 {
        vlan-id 300;
    }
    bd500 {
        vlan-id 500;
    }
    bd600 {
        vlan-id 600;
    }
}
}

```

Configuring ERP on AS1

CLI Quick Configuration To quickly configure AS1 for ERP, copy the following commands and paste them into the switch terminal window of AS1:

```

[edit]
set interfaces ge-2/0/5 unit 0 family bridge interface-mode trunk
set interfaces ge-2/0/5 unit 0 family bridge vlan-id-list 100-1000
set interfaces ge-2/1/1 unit 0 family bridge interface-mode trunk
set interfaces ge-2/1/1 unit 0 family bridge vlan-id-list 100-1000
set protocols protection-group ethernet-ring ring-1 east-interface control-channel ge-2/0/5.0
set protocols protection-group ethernet-ring ring-1 east-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 west-interface control-channel ge-2/1/1.0
set protocols protection-group ethernet-ring ring-1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring ring-1 data-channel vlan [200, 300]
set protocols protection-group ethernet-ring ring-2 east-interface control-channel ge-2/0/5.0
set protocols protection-group ethernet-ring ring-2 east-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 west-interface control-channel ge-2/1/1.0
set protocols protection-group ethernet-ring ring-2 west-interface control-channel vlan 101
set protocols protection-group ethernet-ring ring-2 data-channel vlan [500, 600]
set bridge-domains bd100 vlan-id 100
set bridge-domains bd101 vlan-id 101
set bridge-domains bd200 vlan-id 200
set bridge-domains bd300 vlan-id 300
set bridge-domains bd500 vlan-id 500
set bridge-domains bd600 vlan-id 600

```

Step-by-Step Procedure To configure ERP on AS1:

1. Configure the trunk interface **ge-2/0/5** to connect CS2 to CS1 and trunk interface **ge-2/1/1** to connect CS2 to CS1, and configure the **family** statement as **bridge** with a VLAN ID list of 100 through 1000:

```

[edit interfaces]
user@as1# set ge-2/0/5 unit 0 family bridge interface-mode trunk
user@as1# set ge-2/0/5 unit 0 family bridge vlan-id-list 100-1000
user@as1# set ge-2/1/1 unit 0 family bridge interface-mode trunk
user@as1# set ge-2/1/1 unit 0 family bridge vlan-id-list 100

```

2. Enable ERP, specifying the control channels and data channels for **ring-1** and **ring-2**:



NOTE: Always configure the east-interface statement first, before configuring the west-interface statement.

- ```
[edit protection-group]
user@as1# set ethernet-ring ring-1 east-interface control-channel ge-2/0/5.0
user@as1# set ethernet-ring ring-1 east-interface control-channel vlan 100
user@as1# set ethernet-ring ring-1 west-interface control-channel ge-2/1/1.0
user@as1# set ethernet-ring ring-1 west-interface control-channel vlan 100
user@as1# set ethernet-ring ring-2 east-interface control-channel ge-2/0/5.
user@as1# set ethernet-ring ring-2 east-interface control-channel vlan 101
user@as1# set ethernet-ring ring-2 west-interface control-channel ge-2/1/1.0
user@as1# set ethernet-ring ring-2 west-interface control-channel vlan 101
user@as1# set ethernet-ring ring-2 data-channel vlan [500, 600]
```
3. Configure the routing instance, the bridge domains, and the VLAN IDs associated with each bridge domain:

```
[edit bridge-domains]
user@as1# set bd100 vlan-id 100
user@as1# set bd101 vlan-id 101
user@as1# set bd200 vlan-id 200
user@as1# set bd300 vlan-id 300
user@as1# set bd500 vlan-id 500
user@as1# set bd600 vlan-id 600
```

**Results** Check the results of the configuration:

```
user@as1> show configuration
interfaces {
 ge-2/0/5 {
 unit 0 {
 family bridge {
 interface-mode trunk;
 vlan-id-list 100-1000
 }
 }
 }
 ge-2/1/1 {
 unit 0 {
 family bridge {
 interface-mode trunk;
 vlan-id-list 100-1000
 }
 }
 }
}
protocols {
 protection-group {
 ethernet-ring ring-1 {
 east-interface {
 control-channel {
 ge-2/0/5.0;
 vlan 100;
 }
 }
 west-interface {
 control-channel {
 ge-2/1/1.0;
 vlan 100;
 }
 }
 data-channel {
```

```
 vlan [200, 300];
 }
}
}
protection-group {
 ethernet-ring ring-2 {
 east-interface {
 control-channel {
 ge-2/0/5.0;
 vlan 101;
 }
 }
 west-interface {
 control-channel {
 ge-2/1/1.0;
 vlan 101;
 }
 }
 data-channel {
 vlan [500, 600];
 }
 }
}
bridge-domains {
 bd100 {
 vlan-id 100;
 }
 bd101 {
 vlan-id 101;
 }
 bd200 {
 vlan-id 200;
 }
 bd300 {
 vlan-id 300;
 }
 bd500 {
 vlan-id 500;
 }
 bd600 {
 vlan-id 600;
 }
}
}
```

## Verification

To confirm that the ERP configuration for multiple ring instances is operating, perform these tasks:

- [Verifying the Ethernet Protection Ring on CS1 on page 173](#)
- [Verifying the Data Channel CS1 on page 173](#)
- [Verifying the VLANs on CS1 on page 174](#)



- [Verifying the Ethernet Protection Ring on CS2 on page 174](#)
- [Verifying the Data Channel CS2 on page 175](#)
- [Verifying the VLANs on CS2 on page 175](#)
- [Verifying the Ethernet Protection Ring on AS1 on page 176](#)
- [Verifying the Data Channels on AS1 on page 176](#)
- [Verifying the VLANs on AS1 on page 177](#)

### Verifying the Ethernet Protection Ring on CS1

**Purpose** Verify that ERP is enabled on CS1.

**Action** Show the status of the ring automatic protection switching (R-APS) messages to determine if there is a ring failure:

```
user@cs1>show protection-group ethernet-ring aps
```

| Ethernet<br>Node ID | Ring Name | Request/state | No Flush | Ring Protection | Link Blocked | Originator | Remote |
|---------------------|-----------|---------------|----------|-----------------|--------------|------------|--------|
| ring-1              |           | NR            | No       | Yes             | Yes          | Yes        |        |
| ring-2              |           | NR            | No       | Yes             | Yes          | No         |        |
| 13:22:af:31:fc:00   |           |               |          |                 |              |            |        |

**Meaning** The output displayed shows that protection groups **ring-1** and **ring-2** have a **Request/state** of **NR**, meaning there is no request for APS on the ring. If a **Request/state** of **SF** is displayed, it indicates there is a signal failure on the ring. The output also shows that the ring protection link is not blocked. The **No Flush** field displays **No**, indicating that MAC addresses will be flushed when the ring nodes receive this message first time. A value of **Yes** would indicate MAC address flushing is not needed. The **Originator** field for **ring-1** displays **yes**, indicating that this node is an R-APS originator.

### Verifying the Data Channel CS1

**Purpose** Verify the forwarding state of the data channel.

**Action** List the interfaces acting as the control channels and their respective data channels (represented by the Spanning Tree Protocol (STP) index number):

```
user@cs1>show protection-group ethernet-ring data-channel
```

```
Ethernet ring data channel parameters for protection group ring-1
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
| ge-3/2/4  | 122       | forwarding    |
| ge-5/2/3  | 123       | forwarding    |

```
Ethernet ring data channel parameters for protection group ring-2
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
|-----------|-----------|---------------|

|          |     |            |
|----------|-----|------------|
| ge-3/2/4 | 124 | discarding |
| ge-5/2/3 | 125 | forwarding |

**Meaning** The output displayed shows the STP index number used by each interface in ring instances **ring-1** and **ring-2**. The STP index controls the forwarding behavior for a set of VLANs on the data channel of a ring instance on a ring interface. For ring instances, there are multiple STP index numbers (here representing VLANs 200, 300, 500, and 600). The **Forward State** shows whether the data channel is **forwarding** or **discarding** traffic.

### Verifying the VLANs on CS1

**Purpose** Verify the data channel logical interfaces and the VLAN IDs controlled by a ring instance data channel.

**Action** List dynamic VLAN membership:

```
user@cs1>show protection-group ethernet-ring vlan
Ethernet ring IFBD parameters for protection group ring-1
```

| Interface | Vlan | STP Index | Bridge Domain |
|-----------|------|-----------|---------------|
| ge-3/2/4  | 200  | 122       | vs/bd200      |
| ge-5/2/3  | 200  | 123       | vs/bd200      |
| ge-3/2/4  | 300  | 122       | vs/bd300      |
| ge-5/2/3  | 300  | 123       | vs/bd300      |

```
Ethernet ring IFBD parameters for protection group ring-2
```

| Interface | Vlan | STP Index | Bridge Domain |
|-----------|------|-----------|---------------|
| ge-3/2/4  | 500  | 124       | vs/bd500      |
| ge-5/2/3  | 500  | 125       | vs/bd500      |
| ge-3/2/4  | 600  | 124       | vs/bd600      |
| ge-5/2/3  | 600  | 125       | vs/bd600      |

**Meaning** The output displayed shows the ring interfaces **ge-3/2/4** and **ge-5/2/3** in protection groups **ring-1** and **ring-2**. For **ring-1**, VLAN 200 and VLAN 300 are being supported on both **STP Index 122** and **123** on bridge domains **bd200** and **bd300**. For **ring-2**, VLAN 500 and VLAN 600 are being supported on both **STP Index 124** and **125** on bridge domains **bd500** and **bd600**. The data channel controls the traffic on the VLAN IDs to facilitate load balancing.

### Verifying the Ethernet Protection Ring on CS2

**Purpose** Verify that ERP is enabled on CS2.

**Action** Show the status of the ring APS (R-APS) messages to determine if there is a ring failure:

```
user@cs2>show protection-group ethernet-ring aps
```

| Ethernet Ring Name<br>Node ID | Request/state | No Flush | Ring Protection    | Originator | Remote |
|-------------------------------|---------------|----------|--------------------|------------|--------|
| Ring-1<br>00:21:59:03:ff:d0   | NR            | No       | Link Blocked<br>No | No         |        |
| Ring-2                        | NR            | No       | Yes                | Yes        |        |

**Meaning** The output displayed shows that protection groups **ring-1** and **ring-2** have a **Request/state** of **NR**, meaning there is no request for APS on the ring. If a **Request/state** of **SF** is displayed, it indicates there is a signal failure on the ring. The output also shows that the ring protection link is not blocked. The **No Flush** field displays **No**, indicating that MAC addresses will be flushed when the ring nodes receive this message first time. A value of **Yes** would indicate MAC address flushing is not needed. The **Originator** field for **ring-1** displays **yes**, indicating that this node is an R-APS originator. The **Originator** field for **ring-2** displays **No**, indicating that this node is not an R-APS originator.

### Verifying the Data Channel CS2

**Purpose** Verify the forwarding state of the data channel.

**Action** List the interfaces acting as the control channels and their respective data channels (represented by the STP index number):

```
user@cs2> show protection-group ethernet-ring data-channel
Ethernet ring data channel parameters for protection group ring-1
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
| ge-2/0/4  | 44        | forwarding    |
| ge-2/0/8  | 45        | forwarding    |

```
Ethernet ring data channel parameters for protection group ring-2
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
| ge-2/0/4  | 46        | forwarding    |
| ge-2/0/8  | 47        | discarding    |

**Meaning** The output displayed shows the STP index number used by each interface in ring instances **ring-1** and **ring-2**. The STP index controls the forwarding behavior for a set of VLANs on the data channel of a ring instance on a ring interface. For ring instances, there are multiple STP index numbers (here representing VLANs 200, 300, 500, and 600). The **Forward State** shows whether the data channel is **forwarding** or **discarding** traffic.

### Verifying the VLANs on CS2

**Purpose** Verify the data channel logical interfaces and the VLAN IDs controlled by a ring instance data channel.

**Action** List dynamic VLAN membership:

```
user@cs2> show protection-group ethernet-ring vlan
Ethernet ring IFBD parameters for protection group ring-1
```

| Interface | Vlan | STP Index | Bridge Domain        |
|-----------|------|-----------|----------------------|
| ge-2/0/4  | 200  | 44        | default-switch/bd200 |
| ge-2/0/8  | 200  | 45        | default-switch/bd200 |
| ge-2/0/4  | 300  | 44        | default-switch/bd300 |
| ge-2/0/8  | 300  | 45        | default-switch/bd300 |

```
Ethernet ring IFBD parameters for protection group ring-2
```

| Interface | Vlan | STP Index | Bridge Domain        |
|-----------|------|-----------|----------------------|
| ge-2/0/4  | 500  | 46        | default-switch/bd500 |
| ge-2/0/8  | 500  | 47        | default-switch/bd500 |
| ge-2/0/4  | 600  | 46        | default-switch/bd600 |
| ge-2/0/8  | 600  | 47        | default-switch/bd600 |

**Meaning** The output displayed shows the ring interfaces **ge-2/0/4** and **ge-2/0/8** in protection groups **ring-1** and **ring-2**. For **ring-1**, VLAN 200 and VLAN 300 are being supported on both **STP Index 44** and **45** on bridge domains **bd200** and **bd300**. For **ring-2**, VLAN 500 and VLAN 600 are being supported on both **STP Index 46** and **47** on bridge domains **bd500** and **bd600**. The data channel controls the traffic on the VLAN IDs to facilitate load balancing.

### Verifying the Ethernet Protection Ring on AS1

**Purpose** Verify that ERP is enabled on AS1.

**Action** Show the status of the ring APS (R-APS) messages to determine if there is a ring failure:

```
user@as1> show protection-group ethernet-ring aps
```

| Ethernet Ring Name<br>Node ID | Request/state | No Flush | Ring Protection     | Originator | Remote |
|-------------------------------|---------------|----------|---------------------|------------|--------|
| Ring-1<br>00:21:59:03:ff:d0   | NR            | No       | Link Blocked<br>Yes | No         |        |
| Ring-2<br>13:22:af:31:fc:00   | NR            | No       | Yes                 | No         |        |

**Meaning** The output displayed shows that protection groups **ring-1** and **ring-2** have a **Request/state** of **NR**, meaning there is no request for APS on the ring. If a **Request/state** of **SF** is displayed, it indicates there is a signal failure on the ring. The output also shows that the ring protection link is not blocked. The **No Flush** field displays **No**, indicating that MAC addresses will be flushed when the ring nodes receive this message first time. A value of **Yes** would indicate MAC address flushing is not needed. The **Originator** field for **ring-1** and **ring-2** displays **No**, indicating that this node is not the R-APS originator.

### Verifying the Data Channels on AS1

**Purpose** Verify the forwarding state of the data channel.

**Action** List the interfaces acting as the control channels and their respective data channels (represented by the STP index number):

```
user@as1> show protection-group ethernet-ring data-channel
Ethernet ring data channel parameters for protection group ring-1
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
| ge-2/0/5  | 22        | forwarding    |
| ge-2/1/1  | 23        | forwarding    |

```
Ethernet ring data channel parameters for protection group ring-2
```

| Interface | STP index | Forward State |
|-----------|-----------|---------------|
| ge-2/0/5  | 24        | forwarding    |
| ge-2/1/1  | 25        | forwarding    |

**Meaning** The output displayed shows the STP index number used by each interface in ring instances **ring-1** and **ring-2**. The STP index controls the forwarding behavior for a set of VLANs on the data channel of a ring instance on a ring interface. For ring instances, there are multiple STP index numbers (here representing VLANs 200, 300, 500, and 600). The **Forward State** shows whether the data channel is **forwarding** or **discarding** traffic. All data channels are forwarding traffic.

### Verifying the VLANs on AS1

**Purpose** Verify the data channel logical interfaces and the VLAN IDs controlled by a ring instance data channel.

**Action** List dynamic VLAN membership:

```
user@as1>show protection-group ethernet-ring vlan
```

Ethernet ring IFBD parameters for protection group ring-1

| Interface | Vlan | STP Index | Bridge Domain        |
|-----------|------|-----------|----------------------|
| ge-2/0/5  | 200  | 22        | default-switch/bd200 |
| ge-2/1/1  | 200  | 23        | default-switch/bd200 |
| ge-2/0/5  | 300  | 22        | default-switch/bd300 |
| ge-2/1/1  | 300  | 23        | default-switch/bd300 |

Ethernet ring IFBD parameters for protection group ring-2

| Interface | Vlan | STP Index | Bridge Domain        |
|-----------|------|-----------|----------------------|
| ge-2/0/5  | 500  | 24        | default-switch/bd500 |
| ge-2/1/1  | 500  | 25        | default-switch/bd500 |
| ge-2/0/5  | 600  | 24        | default-switch/bd600 |
| ge-2/1/1  | 600  | 25        | default-switch/bd600 |

**Meaning** The output displayed shows the ring interfaces **ge-2/0/5** and **ge-2/1/1** in protection groups **ring-1** and **ring-2**. For **ring-1**, VLAN 200 and VLAN 300 are being supported on both **STP Index 22** and **23** on bridge domains **bd200** and **bd300**. For **ring-2**, VLAN 500 and VLAN 600 are being supported on both **STP Index 24** and **25** on bridge domains **bd500** and **bd600**. The data channel controls the traffic on the VLAN IDs to facilitate load-balancing.

**Related Documentation**

- [Ethernet Ring Protection Using Ring Instances for Load Balancing on page 153](#)
- [Ethernet Ring Protection on page 151](#)

## Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation

Under normal operating conditions, when Ethernet ring protection is configured correctly, the ring protection link (RPL) owner (Router 1 in the configuration example) will see the following:

**Router 1 Operational  
Commands (Normal  
Ring Operation)**

```
user@router1> show protection-group ethernet-ring aps
Ethernet Ring Name Request/state No Flush Ring Protection Link Blocked
pg101 NR No Yes
```

```
Originator Remote Node ID
Yes
```

Note that the ring protection link is blocked and the node is marked as the originator of the protection.

```
user@router1> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg101
```

| Interface | Control Channel | Forward State | Ring Protection Link End |
|-----------|-----------------|---------------|--------------------------|
| ge-1/0/1  | ge-1/0/1.1      | discarding    | Yes                      |
| ge-1/2/4  | ge-1/2/4.1      | forwarding    | No                       |

| Signal Failure | Admin State |
|----------------|-------------|
| Clear          | IFF ready   |
| Clear          | IFF ready   |

Note that the protection interface is discarding while the other interface is forwarding.

```
user@router1> show protection-group ethernet-ring node-state
Ethernet ring APS State Event Ring Protection Link Owner
pg101 idle NR-RB Yes
```

```
Restore Timer Quard Timer Operation state
disabled disabled operational
```

Note that Router 1 is the owner and timers are disabled.

```
user@router1> show protection-group ethernet-ring statistics group-name pg101
```

```
Ethernet Ring statistics for PG pg101
```

|                       |     |
|-----------------------|-----|
| RAPS sent             | : 1 |
| RAPS received         | : 0 |
| Local SF happened:    | : 0 |
| Remote SF happened:   | : 0 |
| NR event happened:    | : 0 |
| NR-RB event happened: | : 1 |

Note that only minimal RAPS messages have been sent to establish the ring.

Under normal operating conditions, the other routers on the ring (Router 2 and Router 3) will see the following similar output:

**Router 2 and Router 3  
Operational  
Commands (Normal  
Ring Operation)**

```
user@router2> show protection-group ethernet-ring aps
Ethernet Ring Name Request/state No Flush Ring Protection Link Blocked
pg102 NR No Yes
```

```
Originator Remote Node ID
No 00:01:01:00:00:01
```

Router 3 will see almost identical information.

```
user@router2> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg102
```

| Interface | Control Channel | Forward State | Ring Protection Link End |
|-----------|-----------------|---------------|--------------------------|
| ge-1/2/1  | ge-1/2/1.1      | forwarding    | No                       |
| ge-1/0/2  | ge-1/0/2.1      | forwarding    | No                       |

| Signal Failure | Admin State |
|----------------|-------------|
| Clear          | IFF ready   |
| Clear          | IFF ready   |

Note that both interfaces are forwarding. Router 3 will see almost identical information.

```
user@router2> show protection-group ethernet-ring node-state
Ethernet ring APS State Event Ring Protection Link Owner
pg102 idle NR-RB No

Restore Timer Quard Timer Operation state
disabled disabled operational
```

Note that Router 2 is not the owner. Router 3 will see almost identical information.

```
user@router2> show protection-group ethernet-ring statistics group-name pg102
Ethernet Ring statistics for PG pg102
RAPS sent : 0
RAPS received : 1
Local SF happened: : 0
Remote SF happened: : 0
NR event happened: : 0
NR-RB event happened: : 1
```

Router 3 will see almost identical information.

#### Related Documentation

- [Ethernet OAM](#)
- [Ethernet Ring Protection on page 151](#)
- [Example: Configuring Ethernet Ring Protection for MX Series Routers on page 154](#)
- [Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition on page 179](#)

## Example: Viewing Ethernet Ring Protection Status—Ring Failure Condition

This section assumes that Ethernet ring protection is configuring correctly, that Router 1 is the ring protection link (RPL) owner, and that there is a link failure between Router 2 and Router 3 in the configuration example.

#### Router 1 Operational Commands (Ring Failure Condition)

```
user@router1> show protection-group ethernet-ring aps
Ethernet Ring Name Request/state No Flush Ring Protection Link Blocked
pg101 SF NO No

Originator Remote Node ID
No 00:01:02:00:00:01
```

Note that the ring protection link is no longer blocked and the node is no longer marked as originator.

```
user@router1> show protection-group ethernet-ring interface
```

Ethernet ring port parameters for protection group pg101

| Interface | Control Channel | Forward State | Ring Protection Link End |
|-----------|-----------------|---------------|--------------------------|
| ge-1/0/1  | ge-1/0/1.1      | forwarding    | Yes                      |
| ge-1/2/4  | ge-1/2/4.1      | forwarding    | No                       |

| Signal Failure | Admin State |
|----------------|-------------|
| Clear          | IFF ready   |
| Clear          | IFF ready   |

Note that the protection interface is now forwarding (so is the other interface).

user@router1> **show protection-group ethernet-ring node-state**

how protection-group ethernet-ring node-state

| Ethernet ring | APS State | Event | Ring Protection Link Owner |
|---------------|-----------|-------|----------------------------|
| pg101         | protected | SF    | Yes                        |

| Restore Timer | Quard Timer | Operation state |
|---------------|-------------|-----------------|
| disabled      | disabled    | operational     |

Note that Router 1 has recorded the span failure (SF).

user@router1> **show protection-group ethernet-ring statistics group-name pg101**

Ethernet Ring statistics for PG pg101

|                       |     |
|-----------------------|-----|
| RAPS sent             | : 1 |
| RAPS received         | : 1 |
| Local SF happened:    | : 0 |
| Remote SF happened:   | : 1 |
| NR event happened:    | : 0 |
| NR-RB event happened: | : 1 |

Note that the R-APS messages have recorded the remote failure.

Under a failure condition, the other routers on the ring (Router 2 and Router 3) will see the following similar output:

#### Router 2 and Router 3 Operational Commands (Failure Condition)

user@router2> **show protection-group ethernet-ring aps**

| Ethernet Ring Name | Request/state | No Flush | Ring Protection Link Blocked |
|--------------------|---------------|----------|------------------------------|
| pg102              | SF            | No       | No                           |

| Originator | Remote Node ID    |
|------------|-------------------|
| Yes        | 00:00:00:00:00:00 |

Note the failure event (SF). Router 3 will see almost identical information.

user@router2> **show protection-group ethernet-ring interface**

Ethernet ring port parameters for protection group pg102

| Interface | Control Channel | Forward State | Ring Protection Link End |
|-----------|-----------------|---------------|--------------------------|
| ge-1/2/1  | ge-1/2/1.1      | forwarding    | No                       |
| ge-1/0/2  | ge-1/0/2.1      | discarding    | No                       |

| Signal Failure | Admin State |
|----------------|-------------|
| Clear          | IFF ready   |
| set            | IFF ready   |



Note that the failed interface (**ge-1/0/2.1**) is not forwarding. Router 3 will see almost identical information.

```
user@router2> show protection-group ethernet-ring node-state
Ethernet ring APS State Event Ring Protection Link Owner
pg102 idle NR-RB No

Restore Timer Quard Timer Operation state
disabled disabled operational
```

Note that Router 2 is not the owner. Router 3 will see almost identical information.

```
user@router2> show protection-group ethernet-ring statistics group-name pg102
Ethernet Ring statistics for PG pg102
RAPS sent : 1
RAPS received : 1
Local SF happened: : 1
Remote SF happened: : 0
NR event happened: : 0
NR-RB event happened: : 1
```

Note that the R-APS messages have recorded the remote failure. Router 3 will see almost identical information.

#### Related Documentation

- [Ethernet OAM](#)
- [Ethernet Ring Protection on page 151](#)
- [Example: Configuring Ethernet Ring Protection for MX Series Routers on page 154](#)
- [Example: Viewing Ethernet Ring Protection Status—Normal Ring Operation on page 177](#)



## PART 4

# Index

- [Index on page 185](#)



# Index

## Symbols

|                                              |       |
|----------------------------------------------|-------|
| #, comments in configuration statements..... | xviii |
| ( ), in syntax descriptions.....             | xviii |
| < >, in syntax descriptions.....             | xvii  |
| [ ], in configuration statements.....        | xviii |
| { }, in configuration statements.....        | xviii |
| (pipe), in syntax descriptions.....          | xviii |

## A

|                               |        |
|-------------------------------|--------|
| acronyms                      |        |
| Ethernet.....                 | 3      |
| addresses                     |        |
| Layer 2 and Layer 3.....      | 7      |
| ATM interworking              |        |
| MX Series router.....         | 83     |
| MX Series router example..... | 85     |
| automatic bridge domains      |        |
| on MX Series.....             | 63, 64 |

## B

|                                          |        |
|------------------------------------------|--------|
| benefits of Ethernet.....                | 9      |
| braces, in configuration statements..... | xviii  |
| brackets                                 |        |
| angle, in syntax descriptions.....       | xvii   |
| square, in configuration statements..... | xviii  |
| bridge domains                           |        |
| MX Series examples.....                  | 63, 64 |
| bridges                                  |        |
| defined.....                             | 6      |
| bridging                                 |        |
| packet flow.....                         | 44     |
| bulk bridge domain configuration         |        |
| on MX Series.....                        | 63     |

## C

|                                            |       |
|--------------------------------------------|-------|
| CFM                                        |       |
| Ethernet overview.....                     | 110   |
| comments, in configuration statements..... | xviii |
| configuring                                |       |
| Ethernet frame delay.....                  | 128   |
| MX Series bridge domains.....              | 30    |

|                                                |    |
|------------------------------------------------|----|
| MX Series integrated bridging and routing..... | 34 |
| MX Series interfaces and VLAN tags.....        | 24 |
| MX Series Layer 2 basics.....                  | 22 |
| MX Series spanning tree protocols.....         | 32 |

## conventions

|                                                |       |
|------------------------------------------------|-------|
| text and syntax.....                           | xvii  |
| curly braces, in configuration statements..... | xviii |
| customer support.....                          | xviii |
| contacting JTAC.....                           | xviii |

## D

|                                   |            |
|-----------------------------------|------------|
| DHCP relay                        |            |
| MX Series configuration.....      | 80         |
| MX Series overview.....           | 79         |
| MX Series VLAN configuration..... | 81         |
| documentation                     |            |
| comments on.....                  | xviii      |
| domains                           |            |
| implicit creation for VLANs.....  | 47         |
| dynamic profiles                  |            |
| MX Series VPLS examples.....      | 67, 68, 72 |

## E

|                                           |                    |
|-------------------------------------------|--------------------|
| Ethernet                                  |                    |
| acronyms.....                             | 3                  |
| benefits of.....                          | 9                  |
| CFM.....                                  | 111, 118, 122      |
| CFM overview.....                         | 110                |
| frame delay.....                          | 125                |
| LFM example.....                          | 144, 145, 146, 148 |
| MAC addresses.....                        | 10, 11             |
| metro See Metro Ethernet                  |                    |
| OAM overview.....                         | 109                |
| ring protection example.....              | 154                |
| ring protection failure condition.....    | 179                |
| ring protection normal operation.....     | 177                |
| ring protection overview.....             | 151, 153           |
| terminology.....                          | 3                  |
| VLAN tag nesting.....                     | 13                 |
| VLAN tags.....                            | 11                 |
| Ethernet frame delay                      |                    |
| configuring.....                          | 128                |
| examples.....                             | 131, 136, 140      |
| statistics.....                           | 129, 130           |
| examples                                  |                    |
| bridge network with normalized VLANs..... | 47                 |
| Ethernet frame delay.....                 | 131, 136, 140      |
| Ethernet LFM.....                         | 144, 145, 146, 148 |
| Ethernet ring failure condition.....      | 179                |

|                                     |     |
|-------------------------------------|-----|
| Ethernet ring normal operation..... | 177 |
| Ethernet ring protection.....       | 154 |
| single VPLS for several VLANs.....  | 55  |
| VLAN tags with VPLS labels.....     | 51  |

## F

|                               |                    |
|-------------------------------|--------------------|
| features                      |                    |
| MX Series Layer 2 basics..... | 21                 |
| firewall filter example       |                    |
| for MX Series.....            | 102, 104, 106, 107 |
| firewall filters              |                    |
| for MX Series.....            | 101                |
| font conventions.....         | xvii               |
| frame delay                   |                    |
| Ethernet.....                 | 125                |

## G

|               |   |
|---------------|---|
| glossary..... | 3 |
|---------------|---|

## I

|                                              |     |
|----------------------------------------------|-----|
| icons defined, notice.....                   | xvi |
| IEEE 802.1p priority bits, filtering on..... | 106 |
| interface types                              |     |
| Layer 2.....                                 | 14  |
| interfaces                                   |     |
| access.....                                  | 14  |
| trunk.....                                   | 14  |

## L

|                                                        |     |
|--------------------------------------------------------|-----|
| Layer 2 bridging firewall filter                       |     |
| matching by IEEE 802.1p priority bits.....             | 106 |
| matching by packet loss priority (PLP).....            | 107 |
| learned VLAN priority (IEEE 802.1p), filtering on..... | 106 |

## M

|                                                  |        |
|--------------------------------------------------|--------|
| MAC addresses.....                               | 10, 11 |
| manuals                                          |        |
| comments on.....                                 | xviii  |
| Metro Ethernet.....                              | 15     |
| MX Series                                        |        |
| automatic bridge domains.....                    | 63, 64 |
| basic Layer 2 features.....                      | 21     |
| bulk bridge domain configuration.....            | 63     |
| configuring basic Layer 2.....                   | 22     |
| configuring bridge domains.....                  | 30     |
| configuring integrated bridging and routing..... | 34     |
| configuring interfaces and VLAN tags.....        | 24     |

|                                             |                    |
|---------------------------------------------|--------------------|
| configuring spanning-tree protocols.....    | 32                 |
| Ethernet CFM.....                           | 110                |
| Ethernet LFM.....                           | 143                |
| Ethernet OAM.....                           | 109                |
| example configurations.....                 | 47, 51, 55         |
| firewall filter example.....                | 102, 104, 106, 107 |
| firewall filters.....                       | 101                |
| Layer 2 interface types.....                | 14                 |
| VLAN and VPLS configurations.....           | 63, 64             |
| VLAN normalization.....                     | 43, 45             |
| VLAN translation.....                       | 63                 |
| VPLS pseudowires with dynamic profiles..... | 67, 68, 72         |
| MX Series router                            |                    |
| ATM interworking.....                       | 83                 |
| ATM interworking example.....               | 85                 |

## N

|                               |        |
|-------------------------------|--------|
| networking                    |        |
| with bridges and routers..... | 6      |
| normalization                 |        |
| VLAN.....                     | 43, 45 |
| normalized VLAN.....          | 46     |
| translation.....              | 45     |
| notice icons defined.....     | xvi    |

## O

|                        |               |
|------------------------|---------------|
| OAM                    |               |
| Ethernet CFM.....      | 111, 118, 122 |
| Ethernet LFM.....      | 143           |
| Ethernet overview..... | 109           |

## P

|                                               |       |
|-----------------------------------------------|-------|
| packet flow                                   |       |
| bridging.....                                 | 44    |
| packet loss priority (PLP), filtering on..... | 107   |
| parentheses, in syntax descriptions.....      | xviii |

## R

|                        |          |
|------------------------|----------|
| ring protection        |          |
| Ethernet overview..... | 151, 153 |
| routers                |          |
| defined.....           | 6        |

## S

|                                          |          |
|------------------------------------------|----------|
| statistics                               |          |
| Ethernet frame delay.....                | 129, 130 |
| support, technical See technical support |          |
| syntax conventions.....                  | xvii     |

**T**

|                      |       |
|----------------------|-------|
| technical support    |       |
| contacting JTAC..... | xviii |
| terminology          |       |
| Ethernet.....        | 3     |

**U**

|                                                |     |
|------------------------------------------------|-----|
| user priority (IEEE 802.1p), filtering on..... | 106 |
|------------------------------------------------|-----|

**V**

|                                         |            |
|-----------------------------------------|------------|
| virtual switches                        |            |
| configuration.....                      | 40         |
| overview.....                           | 39         |
| VLAN                                    |            |
| translation on MX Series.....           | 63         |
| VLAN tags.....                          | 11         |
| nesting.....                            | 13         |
| VLAN translation                        |            |
| MX Series examples.....                 | 63         |
| VLANs                                   |            |
| and VPLS.....                           | 46         |
| bridge network example.....             | 47         |
| implicit learning domains.....          | 47         |
| MX Series examples.....                 | 63, 64     |
| MX Series examples with VPLS.....       | 47, 51, 55 |
| normalization and translation.....      | 43, 45     |
| normalized.....                         | 46         |
| single VPLS example.....                | 55         |
| translation.....                        | 45         |
| VPLS labels example.....                | 51         |
| VPLS                                    |            |
| MX Series examples.....                 | 63, 64     |
| MX Series VLAN examples.....            | 47, 51, 55 |
| virtual interfaces.....                 | 46         |
| VPLS pseudowires                        |            |
| with dynamic profiles on MX Series..... | 67, 68, 72 |

