JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

OpenFlow User Guide

Published
2020-03-12

JUNOS

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, 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.

Junos® OS OpenFlow User Guide
Copyright © 2020 Juniper Networks, Inc. All rights reserved.

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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation | viii

Documentation and Release Notes | **viii**

Using the Examples in This Manual | **viii**

- Merging a Full Example | **ix**
- Merging a Snippet | **x**

Documentation Conventions | **x**

Documentation Feedback | **xiii**

Requesting Technical Support | **xiii**

- Self-Help Online Tools and Resources | **xiv**
- Creating a Service Request with JTAC | **xiv**

1 **Overview**

OpenFlow Support on Juniper Networks Devices | 17

Understanding Support for OpenFlow on Devices Running Junos OS | 18

- OpenFlow Overview | **18**
- OpenFlow Virtual Switches | **19**
- OpenFlow Interfaces | **19**

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20

- OpenFlow Operation and Support | **20**
- OpenFlow Forwarding Actions | **24**

Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | 26

Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS | 28

Understanding OpenFlow Flows and Filters on Devices Running Junos OS | 29

Understanding How the OpenFlow Destination MAC Address Rewrite Action Works | 31

Understanding OpenFlow Flow Instructions on Devices Running Junos OS | 32

Understanding How the OpenFlow Group Action Works | 33

Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS | 34

- OpenFlow Flow Entry Timer Overview | 34
- Idle Timeout and Hard Timeout | 35
- Purge Flow Timer | 35

Understanding OpenFlow Barrier Messages on Devices Running Junos OS | 36

Understanding OpenFlow Multipart Messages on Devices Running Junos OS | 37

Supported Open Standards | 38

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | 40

OpenFlow v1.0 Compliance Matrix for QFX5100 and EX4600 Switches | 48

OpenFlow v1.0 Compliance Matrix for EX4550 Switches | 56

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66

2 OpenFlow Basic Configuration

Configuring Support for OpenFlow on MX Series Routers | 78

- Configuring the OpenFlow Interfaces | 78
- Configuring the OpenFlow Protocol | 79
- Configuring the OpenFlow Routing Instance | 80

Example: Enabling OpenFlow on MX Series Routers | 81

Configuring Support for OpenFlow on EX9200 Switches | 88

- Configuring the OpenFlow Interfaces | 89
- Configuring the OpenFlow Protocol | 90
- Configuring the OpenFlow Routing Instance | 91

Example: Enabling OpenFlow on EX9200 Switches | 92

Configuring Support for OpenFlow on QFX5100 and EX4600 Switches | 99

- Configuring the OpenFlow Interfaces | 100
- Configuring the OpenFlow Protocol | 100

Example: Enabling OpenFlow on QFX5100 and EX4600 Switches | 102

Configuring Support for OpenFlow on EX4550 Switches | 108

Configuring the OpenFlow Interfaces | 109

Configuring the OpenFlow Protocol | 109

Example: Enabling OpenFlow on EX4550 Switches | 111

3 Configuring OpenFlow Hybrid Interfaces

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | 119

Configuring OpenFlow Hybrid Interfaces on MX Series Routers | 120

Configuring the Hybrid Physical Interface | 121

Configuring the Hybrid Interface Logical Units | 121

Configuring the Non-Hybrid Interfaces | 122

Configuring OpenFlow | 122

Configuring the Virtual Switch Routing Instances | 123

Example: Configuring OpenFlow Hybrid Interfaces on MX Series Routers | 124

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | 134

Configuring the Hybrid Physical Interface | 134

Configuring the Hybrid Interface Logical Units | 135

Configuring the Non-Hybrid Interfaces | 135

Configuring OpenFlow | 136

Configuring the Virtual Switch Routing Instances | 137

Example: Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | 138

4 Configuring OpenFlow Traffic Steering Across MPLS Networks

Understanding OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | 150

Example: OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | 151

5 Configuration Statements

address (Protocols OpenFlow) | 177

controller (Protocols OpenFlow) | 178

default-action (Protocols OpenFlow) | 180

id (Protocols OpenFlow) | 181

interfaces (Protocols OpenFlow) | 182

openflow (Protocols OpenFlow) | 183

port (Protocols OpenFlow) | 185

protocol (Protocols OpenFlow) | 186

purge-flow-timer (Protocols OpenFlow) | 187

role (Protocols OpenFlow) | 188

switch (Protocols OpenFlow) | 189

traceoptions (Protocols OpenFlow) | 191

6 Operational Commands

OpenFlow Operational Mode Commands | 196

show openflow capability | 197

show openflow controller | 205

show openflow filters | 208

show openflow flows | 212

show openflow groups | 218

show openflow interfaces | 222

show openflow statistics flows | 227

show openflow statistics groups | 231

show openflow statistics interfaces | 234

show openflow statistics packet | 238

show openflow statistics queue | 240

show openflow statistics summary | 243

show openflow statistics tables | 246

show openflow summary | 250

show openflow switch | 252

About the Documentation

IN THIS SECTION

- Documentation and Release Notes | viii
- Using the Examples in This Manual | viii
- Documentation Conventions | x
- Documentation Feedback | xiii
- Requesting Technical Support | xiii

OpenFlow is an open standard that enables you to control traffic and run experimental protocols in an existing network using a remote controller. Use this guide to learn about the OpenFlow v1.0 and v1.3.1 features available on Juniper Networks devices and to configure these features.

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at https://www.juniper.net/documentation/.

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

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*

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

Convention	Description	Examples
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

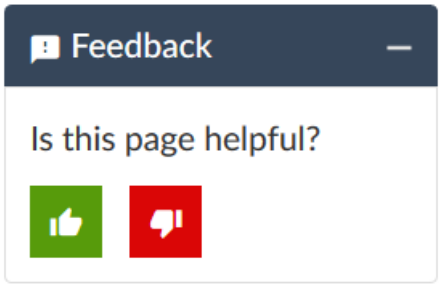
Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

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



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

CHAPTER 1

Overview

OpenFlow Support on Juniper Networks Devices | **17**

Understanding Support for OpenFlow on Devices Running Junos OS | **18**

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | **20**

Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | **26**

Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS | **28**

Understanding OpenFlow Flows and Filters on Devices Running Junos OS | **29**

Understanding How the OpenFlow Destination MAC Address Rewrite Action Works | **31**

Understanding OpenFlow Flow Instructions on Devices Running Junos OS | **32**

Understanding How the OpenFlow Group Action Works | **33**

Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS | **34**

Understanding OpenFlow Barrier Messages on Devices Running Junos OS | **36**

Understanding OpenFlow Multipart Messages on Devices Running Junos OS | **37**

Supported Open Standards | **38**

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | **40**

OpenFlow v1.0 Compliance Matrix for QFX5100 and EX4600 Switches | **48**

OpenFlow v1.0 Compliance Matrix for EX4550 Switches | **56**

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | **66**

OpenFlow Support on Juniper Networks Devices

Starting with Junos OS Release 14.2R1, OpenFlow v1.3.1 support is introduced. The following Juniper Networks devices support OpenFlow v1.0, and OpenFlow v1.3.1:

- EX4600 Switches
- EX9200 Line of Ethernet Switches
- MX80, MX240, MX480, MX960, MX2010, and MX2020 Universal Routing Platforms
- QFX5100 Switches

For these Juniper Networks devices, the OpenFlow software is included in the jsdn package, which is in turn included in the Junos OS software (jinstall) package.

Table 3 on page 17 lists support for various OpenFlow features on Juniper Networks devices that support OpenFlow.

Table 3: OpenFlow Features Supported on Juniper Networks Devices

Juniper Networks Device	Basic OpenFlow Functionality	Hybrid Interfaces	Multi-VLAN Support	OpenFlow over MPLS
EX9200 Line of Ethernet Switches	Yes	Yes	Yes	No
MX80, MX240, MX480, MX960, MX2010, MX2020 Universal Routing Platforms	Yes	Yes	Yes	Yes
QFX5100 Ethernet and EX4600 Switches	Yes	No	Yes	No

Release History Table

Release	Description
14.2R1	Starting with Junos OS Release 14.2R1, OpenFlow v1.3.1 support is introduced.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | 40

Understanding Support for OpenFlow on Devices Running Junos OS

IN THIS SECTION

- OpenFlow Overview | 18
- OpenFlow Virtual Switches | 19
- OpenFlow Interfaces | 19

OpenFlow Overview

OpenFlow is an open standard that enables you to control traffic and run experimental protocols in an existing network by using a remote controller. The OpenFlow components consist of a controller, an OpenFlow or OpenFlow-enabled switch, and the OpenFlow protocol. The OpenFlow protocol is a Layer 2 protocol that permits an OpenFlow controller access to the data plane of an OpenFlow-enabled switch over an SSL or TCP/IP connection.

Using OpenFlow, you can control traffic paths in a network by creating, deleting, and modifying flows in each device along a path. Flow entries specify match conditions against which packets are compared, and a set of actions (OpenFlow v1.0) or instructions (OpenFlow v1.3.1) that are applied to matching packets.

You can configure certain devices running Juniper Networks Junos operating system (Junos OS) as OpenFlow-enabled switches. The Junos OS process, openflowd (ofd), handles OpenFlow functionality on these devices. When implementing OpenFlow in an existing network, you must isolate experimental flows from production flows so that normal network traffic is not impacted. On devices running Junos OS, you isolate OpenFlow traffic by configuring one or more virtual switches that act as logically separate flood domains. The virtual switch and controller communicate by exchanging OpenFlow protocol messages, which the controller uses to add, delete, and modify flows on the switch.

OpenFlow Virtual Switches

To isolate and control OpenFlow traffic on devices running Junos OS, you configure virtual switches. Each virtual switch configuration contains the controller connection information, the set of logical interfaces participating in OpenFlow, and the default action executed when a packet does not match any existing flow entry. You configure the OpenFlow protocol and OpenFlow virtual switches at the **[edit protocols openflow]** hierarchy level.

Depending on the platform, a default VLAN or bridge domain is assigned to each virtual switch. This VLAN or bridge domain acts as a logically separate flood domain, which isolates OpenFlow traffic from normal traffic. On certain platforms, you must also configure a separate virtual switch routing instance at the **[edit routing-instances]** hierarchy level.

You can configure a single OpenFlow virtual switch on devices running Junos OS that support OpenFlow, and you can configure one controller connection per virtual switch. By default, if you configure a virtual switch with a single controller, the controller is in active mode. If a controller is in active mode, the switch automatically initiates a connection to the controller.

OpenFlow Interfaces

When you configure an OpenFlow virtual switch on a device running Junos OS, you must specify the logical interfaces that are participating in OpenFlow for that virtual switch instance. OpenFlow traffic can only either enter or exit OpenFlow-enabled interfaces. MAC address learning is disabled on these interfaces.

Interfaces participating in OpenFlow must be configured as Layer 2 interfaces. To configure an interface as OpenFlow-enabled, you add the logical interface to the OpenFlow virtual switch configuration at the **[edit protocols openflow switch *switch-name* interfaces]** hierarchy level. An OpenFlow interface can be configured only under a single virtual switch. On platforms that require a separate virtual switch routing instance for OpenFlow traffic, you must also configure the OpenFlow interfaces under the OpenFlow virtual switch routing instance.

On certain platforms that support OpenFlow, you can configure only a single logical unit by using logical unit number 0 on an OpenFlow interface. However, on certain platforms that support OpenFlow, a single physical interface can be configured as a hybrid interface that supports both OpenFlow and non-OpenFlow logical interfaces—for example, you can configure interface ge-1/0/1 to have two logical interfaces ge-1/0/1.0 and ge-1/0/1.1, where ge-1/0/1.0 does not participate in OpenFlow, and ge-1/0/1.1 is an OpenFlow-enabled interface.

RELATED DOCUMENTATION

OpenFlow Support on Juniper Networks Devices | 17

Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | 26
Understanding OpenFlow Flows and Filters on Devices Running Junos OS | 29
Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20
OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | 40
OpenFlow v1.0 Compliance Matrix for EX4550 Switches | 56
OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS

IN THIS SECTION

- OpenFlow Operation and Support | 20
- OpenFlow Forwarding Actions | 24

This topic explains how Juniper Networks devices isolate and control OpenFlow traffic. It also summarizes the OpenFlow features and supported forwarding actions, which are actions that OpenFlow can take when a packet matches the terms of a flow entry. For detailed information about support for specific OpenFlow v1.0 messages and fields, match conditions, wildcards, flow actions, statistics, and features, see "OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS" on page 40. For a detailed list of supported OpenFlow v1.3.1 messages and fields, port structure flags and numbering, match conditions, flow actions, multipart messages, flow instructions, and group types, see "OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS" on page 66.

OpenFlow Operation and Support

To isolate and control OpenFlow traffic on devices running Junos OS, you configure virtual switches. You can configure one OpenFlow virtual switch and one active OpenFlow controller on each device running Junos OS that supports OpenFlow. You configure the OpenFlow protocol, virtual switch, and controller connection information at the **[edit protocols openflow]** hierarchy level.

OpenFlow traffic can either enter or exit only OpenFlow-enabled ports. If a flow modification message is sent to an ingress port that is not enabled for OpenFlow, the device sends an ofp_error_msg with an OFPET_FLOW_MOD_FAILED error type and OFPFMFC_UNKNOWN code to the controller. If a flow modification action is requested for a port that is not enabled for OpenFlow, the device sends an ofp_error_msg with an OFPET_BAD_ACTION error type and OFPBAC_BAD_OUT_PORT code to the controller.

Table 4 on page 21 summarizes the general feature support on devices running Junos OS that support OpenFlow v1.0. For information about support on specific platforms, see "OpenFlow Support on Juniper Networks Devices" on page 17.

Table 4: OpenFlow v1.0 Support on Devices Running Junos OS

Feature	Support
OpenFlow v1.0	Supported.
OpenFlow virtual switch	One OpenFlow virtual switch.
Controller	One active OpenFlow controller per virtual switch. Tested controllers include Floodlight and OESS.
Controller connection	TCP/IP connection. Only passive connections are accepted. The controller cannot actively connect to the OpenFlow switch. SSL connections are not supported.
Emergency mode	Not supported as defined in OpenFlow Switch Specification v1.0. If the controller connection is lost and cannot be reestablished, the switch maintains all flow states in the control and data planes.
Flow classification and mapping as a Layer 2 or Layer 3 route	Not supported.
Flow priority	Supported as per OpenFlow Switch Specification v1.3 in which there is no prioritization of exact match entries over wildcard entries.
Flow table	Single flow table.
Forwarding actions	• Forward to an OpenFlow-enabled physical port • ALL, CONTROLLER, NORMAL, and FLOOD for normal flow actions • ALL and FLOOD for Send Packet flow actions **NOTE:** The QFX5100 and EX4600 switches do not support NORMAL for normal flow actions.

Table 4: OpenFlow v1.0 Support on Devices Running Junos OS *(continued)*

Feature	Support
Hybrid interfaces	Supported on some devices. OpenFlow-enabled devices that support hybrid interfaces permit a physical interface to concurrently support logical interfaces for normal traffic and logical interfaces for OpenFlow traffic.
Interfaces	You can configure Ethernet interfaces only as OpenFlow interfaces.
Multi-VLAN actions	Supported on some devices. OpenFlow-enabled devices that support multi-VLAN actions have the ability to associate a different VLAN and different VLAN action with each egress port.
Port modification	Not supported. OpenFlow-enabled devices ignore all OpenFlow controller OFPT_PORT_MOD requests.
Queues, queue messages, or enqueue actions	Not supported.

Table 5 on page 22 summarizes the general feature support on devices running Junos OS that support OpenFlow v1.3.1. For information about support on specific platforms, see "OpenFlow Support on Juniper Networks Devices" on page 17.

Table 5: OpenFlow v1.3.1 Support on Devices Running Junos OS

Feature	Support
OpenFlow v1.3.1	Supported.
OpenFlow virtual switch	One OpenFlow virtual switch.
Controller	One active OpenFlow controller per virtual switch. Tested controllers include NEC and Ixia.
Controller connection	TCP/IP connection. Only passive connections are accepted. The controller cannot actively connect to the OpenFlow switch. SSL connections are not supported.
Flow classification and mapping as a Layer 2 or Layer 3 route	Not supported.

Table 5: OpenFlow v1.3.1 Support on Devices Running Junos OS *(continued)*

Feature	Support
Flow priority	Supported as per OpenFlow Switch Specification v1.3 in which there is no prioritization of exact match entries over wildcard entries.
Flow instructions	For each flow entry, one flow instruction is supported. A flow instruction can be one of the following: • OFPIT_APPLY_ACTIONS • OFPIT_WRITE_ACTIONS
Flow table	Single flow table.
Forwarding actions	• Forward to an OpenFlow-enabled physical port. • ALL, CONTROLLER, NORMAL, and FLOOD for normal flow actions • ALL and FLOOD for Send Packet flow actions **NOTE:** The QFX5100 and EX4600 switches do not support NORMAL for normal flow actions.
Group action	Supported. A group can include 1 through 32 buckets, and a bucket can have a set of actions (set, pop, or output). Group types OFPGT_ALL and OFPGT_INDIRECT are supported.
Interfaces	You can configure Ethernet interfaces only as OpenFlow interfaces.
IPv6-related match conditions	Supported on some devices. Starting with Junos OS Release 14.2R3, IPv6 source and destination addresses and subnet masks can be used as match conditions. **NOTE:** The Junos OS implementation of OpenFlow v1.3.1 does not support arbitrary bit masks for IPv6 addresses. The Junos OS implementation supports only continuous masks for IPv6 source and destination addresses.

Table 5: OpenFlow v1.3.1 Support on Devices Running Junos OS *(continued)*

Feature	Support
Multi-VLAN actions	Supported on some devices. OpenFlow-enabled devices that support multi-VLAN actions have the ability to associate a different VLAN and different VLAN action with each egress port.
Multipart messages	Supported for requesting and returning the following information: • Switch, group, or port descriptions • Single-flow, aggregate-flow, flow table, port, or group statistics • Group or table features
OpenFlow version negotiation	Supported for OpenFlow version negotiation between an OpenFlow controller and a device running Junos OS.
Port modification	Not supported. OpenFlow-enabled devices ignore all OpenFlow controller OFPT_PORT_MOD requests.
Queues, queue messages, or enqueue actions	Not supported.

OpenFlow Forwarding Actions

NOTE: The information in this section applies to both OpenFlow v1.0 and OpenFlow v1.3.1 except where noted.

OpenFlow-enabled devices running Junos OS support several flow actions for forwarding OpenFlow packets. For normal flow actions, the following forwarding actions are supported:

- physical port—Forward unicast or multicast packets out the specified OpenFlow-enabled interfaces.
- ALL—Flood the packet out all OpenFlow interfaces configured for that virtual switch instance except the ingress interface.
- CONTROLLER—Send the packet to the OpenFlow controller for processing.
- FLOOD—Flood the packet along the minimum spanning tree, which includes all OpenFlow interfaces configured for that virtual switch instance except the ingress interface and any interfaces that are disabled

by the Spanning Tree Protocol (STP). Because devices running Junos OS do not support 802.1D STP capabilities for OpenFlow, the FLOOD forwarding action behaves like the ALL forwarding action.

- NORMAL—Process the packet, using traditional Layer 2 or Layer 3 processing.

> **NOTE:** The QFX5100 and EX4600 switches do not support NORMAL for normal flow actions.

The OpenFlow controller can also use a Send Packet message (OFPT_PACKET_OUT) to direct the OpenFlow virtual switch to send a packet out of a specified port. The Send Packet message includes the packet to be forwarded and the forwarding action indicating the interface out of which the packet must be forwarded. Supported forwarding actions for the Send Packet message include ALL and FLOOD.

Each OpenFlow virtual switch is a logically separate flood domain. Therefore, the OpenFlow ALL and FLOOD actions flood packets only out OpenFlow interfaces configured under that specific virtual switch excluding the ingress OpenFlow interface.

Release History Table

Release	Description
14.2R3	Starting with Junos OS Release 14.2R3, IPv6 source and destination addresses and subnet masks can be used as match conditions.

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | 40
- OpenFlow v1.0 Compliance Matrix for EX4550 Switches | 56
- OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66
- Understanding How the OpenFlow Group Action Works | 33
- Understanding OpenFlow Flow Instructions on Devices Running Junos OS | 32
- Understanding OpenFlow Multipart Messages on Devices Running Junos OS | 37
- Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS | 28
- Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | 26
- Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS | 34

Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS

On devices running Juniper Networks Junos operating system (Junos OS), each OpenFlow virtual switch establishes an independent connection with the controller and is represented by a unique runtime datapath ID consisting of the management port MAC address and an internally assigned virtual switch ID. The controller and virtual switch connect to each other using a TCP/IP connection on the management plane. Thus, OpenFlow-enabled devices that are managed by a controller must be connected to the management network (for example, connected using the me0, fxp0, em0, or em1 management port) and must be reachable from the controller IP address.

Upon establishing a connection with the controller, the switch and the controller exchange hello messages that specify the latest OpenFlow protocol version supported by the sender. If the first packet received by the switch is not an OFPT_HELLO message, the switch terminates the connection and attempts to establish a new connection with the controller. Additionally, if the controller and the switch negotiate an OpenFlow protocol version that one of the parties does not support, the connection is terminated with an error message indicating an OFPET_HELLO_FAILED error type and an OFPHFC_INCOMPATIBLE code.

The session is established when the switch and controller successfully exchange Hello messages and negotiate the OpenFlow protocol version. After establishing the session, the controller sends the switch a feature request message requesting the capabilities supported by the switch. The switch responds with a feature reply message, which includes the local MAC address in the virtual switch datapath ID field. If the local MAC address is unavailable, the switch terminates the connection.

After establishing the session, the controller and virtual switch exchange echo request and reply messages as a keepalive mechanism. The keepalive timer is reset if the virtual switch or controller receives either an echo reply or a packet. Echo requests are sent every 10 seconds during idle windows in the absence of other messages. If the switch receives no echo reply or other message from the controller for 120 seconds, the connection is considered lost, and the switch attempts to reestablish the connection for 10 seconds. If the connection cannot be established, the switch enters emergency mode as defined in the OpenFlow v1.3 specification. In emergency mode, the switch deletes normal flow entries, and after 30 seconds, purges flow entries that are installed in hardware.

If at any point after the session is established the recipient receives an OpenFlow message that specifies the wrong OpenFlow version, the recipient responds with an error message indicating an OFPET_BAD_REQUEST type and OFPBRC_BAD_VERSION code. If the switch cannot process the version and type of an OpenFlow packet in the TCP buffer, or if the switch fails sending OpenFlow messages to the controller, the switch terminates the connection.

Modifying, deleting, or deactivating the virtual switch configuration also impacts the connection to the controller. If you modify an existing virtual switch configuration, the virtual switch terminates the existing connection to the controller and establishes a new session with the updated configuration information. If

you delete or deactivate an existing virtual switch configuration, the virtual switch automatically disconnects from the controller.

To summarize, the switch disconnects from the controller under the following circumstances:

- The first packet the switch receives from the controller is not a hello message.
- The switch receives a hello message with an unsupported OpenFlow version.
- The local MAC address is not available for inclusion in the feature reply message.
- The switch receives no echo reply or other message from the controller for 120 seconds.
- The existing virtual switch configuration is deleted or deactivated.
- The existing virtual switch configuration is modified. In this case, after disconnecting from the controller, the switch attempts to establish a new connection and session.
- The switch cannot process the version and type of an OpenFlow packet in the TCP buffer.
- The switch fails to send OpenFlow messages to the controller, which is treated as a dead TCP socket connection.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS | 28

Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS

Upon establishing an initial connection, an OpenFlow controller and a Juniper Networks Junos OS device negotiate the OpenFlow version to be used. In general, the OpenFlow controller must support at least one of the versions run on the Junos OS device. Otherwise, a connection is not established.

> **NOTE:** The Junos OS implementation of OpenFlow 1.3.1 does not support the OFPHET_VERSIONBITMAP Hello message element.

Table 6 on page 28 outlines the OpenFlow versions run by the Junos OS device and controller, the negotiated version, and the *numerical value* associated with each version.

Table 6: OpenFlow Versions Negotiated Between the Controller and a Junos OS Device and the Numerical Value Associated with Each Version

OpenFlow Version Run by Junos OS Device	OpenFlow Version Supported by Controller	Negotiated Version	Numerical Value Associated with Negotiated OpenFlow Version
1.0	1.0	1.0	1
1.3.1	1.3.1	1.3.1	4
1.0 and 1.3.1	1.0 and 1.3.1	1.3.1	4
1.0 and 1.3.1	1.0	1.0	1
1.0 and 1.3.1	1.3.1	1.3.1	4
1.0 and/or 1.3.1	• Neither 1.0 nor 1.3.1 • Connection with Junos OS device is down	None; no connection	0

To determine the negotiated version running on a Junos OS device, you enter the **show openflow controller** command. The output of this command includes a **Negotiated version** field and a numerical value that

represents the negotiated version number. Use Table 6 on page 28 to correlate the numerical values shown in this field with the negotiated versions.

RELATED DOCUMENTATION

Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | 26

Understanding OpenFlow Flows and Filters on Devices Running Junos OS

OpenFlow flows are defined by various elements. Table 7 on page 29 outlines the support for flow elements in OpenFlow v1.0 and OpenFlow v1.3.1. The elements supported by the OpenFlow versions uniquely identify a flow.

Table 7: OpenFlow Flow Elements

Flow Element	Supported In OpenFlow v1.0?	Supported In OpenFlow v1.3.1?
Match conditions	Yes	Yes
Set of actions	Yes	No
Flow instructions	No	Yes
Flow priority	Yes	Yes
Flow timeout information	Yes	Yes
Flow cookie and cookie mask	No	Yes

Flow entries specify wildcard match conditions for fields that do not require an exact match. If a flow entry contains wildcards for all match conditions, then all packets match that flow entry.

To implement OpenFlow flow-based forwarding, devices running Junos OS use filters. For each logical interface configured to participate in OpenFlow, a single filter is created and applied to the logical interface in the input direction. The filter name is the concatenation of the interface name, including the logical unit number, and an internally assigned virtual switch ID, for example ge-1/1/0.0_0.

> **NOTE:** If you manually configure a filter name or a filter term name that conflicts with an autogenerated OpenFlow filter name or filter term name, Junos OS does not generate an error during a **commit check**. If there is a conflict, the commit succeeds, but one of the filters or filter terms is rejected based on the order in which they were received.

A filter consists of one or more terms with match conditions, and actions (for OpenFlow v1.0) or instructions (for OpenFlow v1.3.1). OpenFlow flows are mapped to filter terms, and OpenFlow controller requests to add, delete, and modify flows result in the addition, deletion, or modification of terms in the filter. When the OpenFlow controller sends a flow modification request, the flow entry match condition for the ingress port determines which logical interface filter is updated. The OpenFlow flow priority determines the order of the terms in the filter, where higher priority terms are installed above lower priority terms. Flow match conditions are mapped to the filter term match conditions, and flow actions or instructions are mapped to the filter term **then** statement. Depending on the flow action or instruction, the **then** statement might include actions for forwarding the packet to the next hop or OpenFlow controller, or discarding the packet.

> **NOTE:** If the OpenFlow controller sends a request to modify a flow, but no flow entries match the conditions, OpenFlow v1.0 adds an entry for the flow to the flow table. However, in the same situation, OpenFlow v1.3.1 does not add this flow to the flow table, nor is an error logged.

Each OpenFlow flow entry corresponds to a filter term. However, each flow entry might map to a term in one or more filters depending on the match condition for the ingress port. If the ingress port is a wildcard match, the flow entry appears as a term in all of the interface filters for that OpenFlow virtual switch. For example, suppose that the OpenFlow controller sends a request to add a new flow entry with a wildcard match for the ingress port field. In this case, the flow is added as a new filter term for all OpenFlow logical interfaces configured under that virtual switch.

Devices running Junos OS support both strict and non-strict flow mod commands for modifying and deleting flows. OpenFlow controller strict-modify and strict-delete flow mod requests modify or delete only flows that exactly match the description for all header fields including wildcards and priorities. Non-strict modify and delete flow mod requests modify or delete flows that exactly match or are more specific than the request.

In addition to the functionality already described, OpenFlow v1.3.1 supports a flow cookie, which is an identifier that the OpenFlow controller can specify when a flow is installed in the flow table. This cookie enables OpenFlow to filter flows selected for flow modification and delete operations.

You can configure the default action for packets that do not match on any flow entry as either **drop**, which discards the packet, or **packet-in**, which accepts the packet and forwards it to the controller. The default action is specific to the OpenFlow virtual switch and is the same across all filters associated with that virtual switch. If you do not explicitly configure the default action, the default is **packet-in**.

In the event that a logical interface becomes unavailable, the filter associated with that logical interface is removed from the Packet Forwarding Engine. Although the filter is removed, the Routing Engine retains flows that match the logical interface as the ingress port until such time as the flows are purged in response to OpenFlow timers. For information about OpenFlow timers, see "Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS" on page 34. If the logical interface becomes available before the flows are purged, the filter and any flows retained by the Routing Engine at that point are reinstalled in hardware.

Similarly, when a logical interface becomes unavailable, flows that have that logical interface as the only active egress interface in their action set or instruction are considered invalid. The invalid flows are removed from the Packet Forwarding Engine but are indefinitely retained by the Routing Engine until the flows are purged in response to various OpenFlow timers. Alternatively, flows that include the logical interface as one of several active egress interfaces in their action set or instruction are still valid. In that case, the flow remains in the Packet Forwarding Engine, but the multicast next hop is updated to remove that logical interface as a valid egress interface.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Understanding OpenFlow Flow Instructions on Devices Running Junos OS | 32

Understanding How the OpenFlow Destination MAC Address Rewrite Action Works

Some types of network equipment that function as routers accept and handle packets only if the destination MAC address in the packet is the same as the MAC address of the Layer 3 interface on which the packet is received. To interoperate with these routers, connected devices must also be able to rewrite the destination MAC address of an incoming packet. Starting with Junos OS Releases 14.2R6, an OpenFlow controller can configure an MX Series router that supports OpenFlow to rewrite the destination MAC address of an incoming packet.

The MX routers support a maximum of two actions in a flow. As a result, the MX routers support the following flow combinations:

- Destination MAC address rewrite and VLAN SWAP
- Destination MAC address rewrite and VLAN POP
- Destination MAC address rewrite in a group

If a flow includes an unsupported combination of actions, for example, VLAN PUSH and VLAN POP, the MX Series routers reject the flow.

Release History Table

Release	Description
14.2R6	Starting with Junos OS Releases 14.2R6, an OpenFlow controller can configure an MX Series router that supports OpenFlow to rewrite the destination MAC address of an incoming packet.

Understanding OpenFlow Flow Instructions on Devices Running Junos OS

NOTE: Flow instructions are supported only on Juniper Networks devices running OpenFlow v1.3.1 or later.

When a packet matches a particular OpenFlow flow, a Juniper Networks device running OpenFlow v1.0 applies a set of actions to the packet. Starting with OpenFlow v1.3.1, instead of applying a set of actions, the Juniper Networks device applies a flow instruction to a matching packet.

In the Junos OS implementation of OpenFlow v1.3.1, a flow entry can include only one flow instruction, which can be one of the following:

- Apply actions (OFPIT_APPLY_ACTIONS)
- Write actions (OFPIT_WRITE_ACTIONS)

Each of the instructions mentioned above includes a list of actions that the device applies immediately in the order in which they appear the list.

RELATED DOCUMENTATION

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66

Understanding How the OpenFlow Group Action Works

> **NOTE:** The group action is supported only on Juniper Networks devices running OpenFlow v1.3.1 or later.

OpenFlow uses flow entries as a means to match flows and specify an action for incoming packets on logical OpenFlow interfaces. The action specified in one or more flow entries can direct packets to, or reference, a base action called a *group* action. The purpose of the group action is to further process these packets and assign a more specific forwarding action to them.

A group can include 1 to 32 buckets, and in turn, a bucket can have a set of actions (set, pop, or output).

For information about the specific actions that are supported for each base type, see the "OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS" on page 66.

Juniper Networks Junos operating system (Junos OS) devices support the following *group types*, which define how buckets are implemented:

- All—Multiple buckets are implemented for the handling of multicast and broadcast packets. Each incoming packet is replicated and processed by each bucket in the group.
- Indirect—One bucket is implemented. An indirect group is typically referenced by multiple flow entries, thereby allowing each of these entities to have a centralized action that can be easily updated.

For example, an all group type with a unique OpenFlow-controller-assigned identifier, say, 50 can have two buckets: bucket 1 and bucket 2. The action associated with bucket 1 might be to set the VLAN ID field in the packet to 3022 and to output the packet to an OpenFlow port—for example, 118. The action associated with bucket 2 might be to set the VLAN ID field in the packet to—for example, 2022—and to output the packet to an OpenFlow port—for example, 117.

You can add a group with one or more buckets on the OpenFlow controller, and the controller pushes the group to the Junos OS devices with which it is connected. Each Junos OS device checks to see whether the group already exists. If it does not, the group is added to the group table on the Junos OS devices. After the group is in the group table, you can modify or delete it from the table by way of the OpenFlow controller.

RELATED DOCUMENTATION

show openflow groups | 218

show openflow statistics groups | 231

Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS

IN THIS SECTION

- OpenFlow Flow Entry Timer Overview | 34
- Idle Timeout and Hard Timeout | 35
- Purge Flow Timer | 35

OpenFlow Flow Entry Timer Overview

For each logical interface participating in OpenFlow on a device running Junos OS, a single filter is created and applied to the logical interface in the input direction. OpenFlow flows are mapped to the filter as filter terms. Each flow has a number of timers associated with it, some of which are configured through the OpenFlow controller while others are configured through the Junos OS CLI. OpenFlow flow entry timers include the idle timeout, the hard timeout, and the purge flow timer. Table 8 on page 34 summarizes the various OpenFlow flow timers. EX4550 switches do not support idle timeout.

Table 8: OpenFlow Flow Entry Timers

Timer	Configured Through	Range (Seconds)
Idle timeout	Controller	0, 11 through 65,535
Hard timeout	Controller	0 through 65,535
Purge flow timer	Junos OS CLI by using the **purge-flow-timer** configuration statement	0 through 300

Idle Timeout and Hard Timeout

Each flow entry has an idle timeout and a hard timeout associated with it, both of which are configured through the OpenFlow controller. The idle timeout is the number of seconds after which a flow entry is removed from the flow table and the hardware provided because no packets match it. The hard timeout is the number of seconds after which the flow entry is removed from the flow table and the hardware whether or not packets match it.

If a flow entry has both an idle timer and a hard timer associated with it, the first timer to expire causes the flow entry to be removed. If the idle timer expires first, the flow entry is removed at that point only if there are no matching packets. Otherwise, the flow entry is removed when the hard timer expires.

When the controller sends a flow entry modification message (OFPT_FLOW_MOD) to the switch, it specifies the idle timeout and hard timeout for that flow entry. On devices running Junos OS, the idle timeout value can be 0, or it can range from 11 through 65,535 seconds. If the controller sets the idle timeout to 0, the flow entry does not experience an idle time out. The hard timeout value can range from 0 through 65,535 seconds. If the controller sets the hard timeout to 0, the flow entry does not experience a hard time out. If the controller requests an invalid timeout value, the switch rejects the flow modification message and sends an error message back to the controller.

Purge Flow Timer

On devices running Junos OS, you can configure a purge flow timer, which is the number of seconds after which an invalid OpenFlow flow entry is deleted from the flow table. The **purge-flow-timer** statement is configured through the Junos OS CLI at the **[edit protocols openflow switch *switch-name*]** hierarchy level. The **purge-flow-timer** value is specific to the OpenFlow virtual switch under which it is configured, and it is the same for all flow entries associated with that virtual switch.

If you do not configure the **purge-flow-timer** statement, the device purges invalid flow entries from hardware, but indefinitely retains the corresponding flow entries in the flow table on the Routing Engine. If you configure the **purge-flow-timer** statement, the device purges invalid flow entries from hardware, and after the specified number of seconds, deletes the invalid flow entries from the flow table. Configuring a value of 0 causes the device to immediately delete invalid flow entries from the flow table.

For example, consider the case of an OpenFlow logical interface that becomes temporarily unavailable. When the interface becomes unavailable, flow entries that have the logical interface as the matching ingress interface or as the only active egress interface in their action set (for OpenFlow v1.0) or flow instruction (for OpenFlow v1.3.1) are marked as invalid. Although the logical interface is not available, the flow entries could still be valid. The **purge-flow-timer** configuration statement determines how to handle the flow entries.

In this example, if you do not configure the **purge-flow-timer** statement, then when the logical interface becomes unavailable, the device removes the invalid flow entries from the hardware but indefinitely retains the flow entries in the flow table. If the logical interface later becomes available, the flow entries are reinstalled in the hardware without any controller intervention.

On the other hand, if you configure the **purge-flow-timer** statement, then when the logical interface becomes unavailable, the device removes the flow entries from the hardware, and retains the flow entries in the flow table for the configured number of **purge-flow-timer** seconds. If the interface does not become available and the timer expires, the device deletes the flow entries from the flow table. After the interface comes back up, the OpenFlow controller must send new flow entry modification messages to the OpenFlow switch in order to restore the flow entries to the flow table and to the hardware.

> **NOTE:** By default, if you remove an active OpenFlow logical interface from an existing OpenFlow configuration, flow entries that match on this logical interface as the ingress interface and flow entries that include this logical interface as the only active egress interface in their action list or flow instruction are invalid and are automatically purged from the flow table and from the hardware regardless of whether you configure the **purge-flow-timer** statement.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | **18**

purge-flow-timer | **187**

Understanding OpenFlow Barrier Messages on Devices Running Junos OS

OpenFlow-enabled devices running Juniper Networks Junos operating system (Junos OS) support the OpenFlow protocol controller-to-switch Barrier Request message (OFPT_BARRIER_REQUEST). The OpenFlow controller sends a Barrier Request message to request that the OpenFlow-enabled switch complete processing of all messages sent before the Barrier Request message before processing any messages sent after the Barrier Request message. This ensures that the virtual switch processes all message dependencies and sends all notifications for completed operations before proceeding with new requests.

When the OpenFlow virtual switch receives a Barrier Request message, it queues all subsequent incoming messages, with the exception of echo request and reply messages, until processing of all prior messages is complete. Echo request and reply messages are required to maintain connectivity to the controller.

When the switch completes an operation, it sends a reply message back to the controller. Only after the reply is sent to the controller does the switch mark the message or operation as processed. After the switch completes processing for all operations requested prior to the Barrier Request message, the switch sends a Barrier Reply (OFPT_BARRIER_REPLY) message, which includes the ID of the original request message, to the OpenFlow controller. At that point, the switch resumes processing of the queued messages.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Understanding OpenFlow Flows and Filters on Devices Running Junos OS | 29

Understanding OpenFlow Multipart Messages on Devices Running Junos OS

NOTE: Multipart messages are supported only on Juniper Networks devices running OpenFlow v1.3.1 or later.

To more efficiently process large OpenFlow data responses, OpenFlow v1.3.1 introduces support for multipart messages.

The OpenFlow controller can use a multipart request message to request the following information:

- Switch, group, or port descriptions
- Single-flow, aggregate-flow, flow table, port, or group statistics
- Group or table features

In response, a Juniper Networks device can send one or more multipart response messages wherein each message includes the same request identifier. In addition, each message in the sequence, except the last message, includes a flag that indicates more messages are to follow.

RELATED DOCUMENTATION

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66

Supported Open Standards

Junos OS substantially supports the following open standards:

- *OpenFlow Switch Specification, Version 1.0.0*

  For a detailed list of supported messages and fields, match conditions, wild cards, flow actions, statistics, and features, see "OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS" on page 40.

  The Junos OS implementation of OpenFlow v1.0 differs from the specification in the following ways:

  (The sections of the OpenFlow specification are indicated in the parentheses.)

  - Junos OS supports only the following flow action types (section 5.2.4):
    - OFPAT_OUTPUT—supports OFPP_NORMAL, OFPP_FLOOD, OFPP_ALL, and OFPP_CONTROLLER for normal flow actions, and OFPP_FLOOD and OFPP_ALL for Send Packet flow actions.
    - OFPAT_SET_VLAN_VID—support varies by platform.
    - OFPAT_STRIP_VLAN—support varies by platform
  - Flow priority is supported according to OpenFlow Switch Specification v1.3.0 in which there is no prioritization of exact match entries over wildcard entries.
  - Emergency mode as defined in OpenFlow v1.0 is not supported. If the controller connection is lost and cannot be reestablished, the switch maintains all flow states in the control and data planes.

  The following features are not supported:

  - Encryption through TLS connection (section 4.4)
  - 802.1D Spanning Tree Protocol (sections 4.5 and 5.2.1)
  - OFPP_LOCAL virtual port (section 5.2.1)
  - Physical port features OFPPF_PAUSE and OFPPF_PAUSE_ASYM (section 5.2.1)
  - Queue structures and queue configuration messages (section 5.2.2 and 5.3.4)
  - Flow action types: OFPAT_SET_VLAN_PCP, OFPAT_SET_DL_SRC/DST, OFPAT_SET_NW_SRC/DST/TOS, OFPAT_SET_TP_SRC/DST and OFPAT_ENQUEUE (section 5.2.4)
  - buffer_id for Modify Flow Entry Message, Send Packet Message, and Packet-In Message (sections 5.3.3, 5.3.6, and 5.4.1)
  - Port Modification Message (section 5.3.3)
  - Vendor Statistics (section 5.3.5)
  - Vendor message (section 5.5.4)

- *OpenFlow Switch Specification, Version 1.3.1*

For a detailed list of supported messages and fields, port structure flags and numbering, match conditions, flow actions, multipart messages, flow instructions, and group types, see "OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS" on page 66.

The Junos OS implementation of OpenFlow v1.3.1 differs from the specification in the following ways:

(The sections of the OpenFlow specification are indicated in the parentheses.)

- Junos OS supports only the following flow action types (section 5.12):
  - OFPAT_SET_VLAN_VID
  - OFPAT_POP_VLAN
  - OFPAT_GROUP
- Junos OS supports only the following group types (section 5.6.1):
  - OFPGT_ALL
  - OFPGT_INDIRECT
- Junos OS supports only one flow instruction per flow entry. Further, only the following flow instructions (section A.2.4) are supported:
  - OFPIT_WRITE_ACTIONS
  - OFPIT_APPLY_ACTIONS
- For OFPT_SET_CONFIG (section A.3.2), Junos OS supports only the OFPC_FRAG_NORMAL configuration flag, and the OFPCML_NO_BUFFER setting for the miss_send_len field.
- On MX Series routers, Junos OS supports only the following IPv6-related match conditions (A.2.3.7):
  - OFPXMT_OFB_IPV6_SRC
  - OFPXMT_OFB_IPV6_DST

The following features are not supported:

- Multiple flow tables (section 5)
- Table metadata (section 2)
- Action sets (section 5.10)
- Meter (section 5.7)
- MPLS fields (section 5.12.1)
- MPLS actions (section 5.10 and 5.12)
- Encryption through TLS connection (section 6.3.3)
- Per-port queues (section A.2.2)
- Auxiliary connections (section 6.3.5)

- Multiple virtual switches (section A.3.1)
- IPv6-related set-field actions (5.12)

RELATED DOCUMENTATION

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS | 40

OpenFlow v1.0 Compliance Matrix for QFX5100 and EX4600 Switches | 48

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS | 66

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS

The following tables list the Junos OS support for OpenFlow v1.0 messages and fields, match conditions, wildcards, flow actions, statistics, and features on the indicated platforms:

- Table 9 on page 40 lists the support for message types.
- Table 10 on page 42 lists the support for port structure flags.
- Table 11 on page 44 lists the support for match conditions.
- Table 12 on page 44 lists the support for wildcards.
- Table 13 on page 45 lists the support for flow actions.
- Table 14 on page 46 lists the support for flow actions in Send Packet messages (OFPT_PACKET_OUT).
- Table 15 on page 47 lists the support for statistics.
- Table 16 on page 47 lists the support for features.

Table 9 on page 40 lists the support for OpenFlow v1.0 message types.

Table 9: Junos OS Support for OpenFlow v1.0 Message Types

Section	Specification	MX Series	EX9200
5.1	OFPT_HELLO	Supported	Supported
	OFPT_ERROR	Supported	Supported

Table 9: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	MX Series	EX9200
	OFPT_ECHO_REQUEST	Supported	Supported
	OFPT_ECHO_REPLY	Supported	Supported
	OFPT_VENDOR	Not supported	Not supported
	OFPT_FEATURES_REQUEST	Supported	Supported
	OFPT_FEATURES_REPLY: Datapath ID N_buffers N_tables OFPC_FLOW_STATS OFPC_TABLE_STATS OFPC_PORT_STATS OFPC_STP OFPC_IP_REASM OFPC_QUEUE_STATS OFPC_ARP_MATCH_IP	Supported Supported 0 1 Supported Supported Supported Not supported Not supported Supported Not supported	Supported Supported 0 1 Supported Supported Supported Not supported Not supported Supported Not supported
	OFPT_GET_CONFIG_REQUEST	Supported	Supported
	OFPT_GET_CONFIG_REPLY	Supported	Supported
	OFPT_SET_CONFIG	Supported	Supported
	OFPT_PACKET_IN OFPT_PACKET_IN with buffer_id	Supported Not supported	Supported Not supported
	OFPT_FLOW_REMOVED	Supported	Supported
	OFPT_PORT_STATUS	Supported	Supported
	OFPT_PACKET_OUT OFPT_PACKET_OUT with buffer_id	Supported Not supported	Supported Not supported

Table 9: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	MX Series	EX9200
	OFPT_FLOW_MOD: OFPFC_ADD OFPFC_ADD with OFPFF_CHECK_OVERLAP OFPFC_MODIFY OFPFC_MODIFY_STRICT OFPFC_DELETE OFPFC_DELETE_STRICT OFPT_FLOW_MOD with buffer_id	Supported Supported Supported Supported Supported Supported Supported Not supported	Supported Supported Supported Supported Supported Supported Supported Not supported
	OFPT_PORT_MOD	Not supported	Not supported
	OFPT_STATS_REQUEST	Supported	Supported
	OFPT_STATS_REPLY See Table 15 on page 47	Supported	Supported
	OFPT_BARRIER_REQUEST	Supported	Supported
	OFPT_BARRIER_REPLY	Supported	Supported
	OFPT_QUEUE_GET_CONFIG_REQUEST	Not supported	Not supported
	OFPT_QUEUE_GET_CONFIG_REPLY	Not supported	Not supported

Table 10 on page 42 lists the support for OpenFlow v1.0 port structure flags.

Table 10: Junos OS Support for OpenFlow v1.0 Port Structure Flags

Section	Specification	MX Series	EX9200
5.2.1	OFPPC_PORT_DOWN	Not supported	Not supported
	OFPPC_NO_STP	Not supported	Not supported
	OFPPC_NO_RECV	Not supported	Not supported
	OFPPC_NO_RECV_STP	Not supported	Not supported
	OFPPC_NO_FLOOD	Not supported	Not supported

Table 10: Junos OS Support for OpenFlow v1.0 Port Structure Flags *(continued)*

Section	Specification	MX Series	EX9200
	OFPPC_NO_FWD	Not supported	Not supported
	OFPPC_NO_PACKET_IN	Not supported	Not supported
	OFPPS_LINK_DOWN	Supported	Supported
	OFPPS_STP_LISTEN	Not supported	Not supported
	OFPPS_STP_LEARN	Not supported	Not supported
	OFPPS_STP_FORWARD	Not supported	Not supported
	OFPPS_STP_BLOCK	Not supported	Not supported
	OFPPS_STP_MASK	Not supported	Not supported
	OFPPF_10MB_HD	Supported	Supported
	OFPPF_10MB_FD	Supported	Supported
	OFPPF_100MB_HD	Supported	Supported
	OFPPF_100MB_FD	Supported	Supported
	OFPPF_1GB_HD	Supported	Supported
	OFPPF_1GB_FD	Supported	Supported
	OFPPF_10GB_FD	Supported	Supported
	OFPPF_COPPER	Supported	Supported
	OFPPF_FIBER	Supported	Supported
	OFPPF_AUTONEG	Supported	Supported
	OFPPF_PAUSE	Not supported	Not supported
	OFPPF_PAUSE_ASYM	Not supported	Not supported

Table 11 on page 44 lists the support for OpenFlow v1.0 match conditions.

Table 11: Junos OS Support for OpenFlow v1.0 Match Conditions

Section	Specification	MX Series	EX9200
5.2.3	dl_src (Ethernet source address)	Supported	Supported
	dl_dst (Ethernet destination address)	Supported	Supported
	dl_vlan (Input VLAN ID) **NOTE:** The flow match condition for the VLAN ID must be less than 4096. Otherwise, the flow is not installed. The only exception is VLAN ID 65535, which corresponds to untagged frames.	Supported	Supported
	dl_vlan_pcp (Input VLAN priority) **NOTE:** The flow match condition for the VLAN priority must be in accordance with 802.1p. Otherwise, the flow is not installed.	Supported	Supported
	dl_type (Ethernet frame type)	Supported	Supported
	nw_tos (IP TOS (6 bits DSCP)	Supported	Supported
	nw_proto (IP Protocol or lower 8 bits of ARP opcode)	Supported	Supported
	nw_src (IP source address)	Supported	Supported
	nw_dst (IP destination address)	Supported	Supported
	tp_src (TCP/UDP source port)	Supported	Supported
	tp_dst (TCP/UDP destination port)	Supported	Supported
	Match all 12 tuples or a combination of tuples	Supported	Supported

Table 12 on page 44 lists the support for OpenFlow v1.0 wildcards.

Table 12: Junos OS Support for OpenFlow v1.0 Wildcards

Section	Specification	MX Series	EX9200
5.2.3	OFPFW_IN_PORT	Supported	Supported
	OFPFW_DL_VLAN	Supported	Supported

Table 12: Junos OS Support for OpenFlow v1.0 Wildcards *(continued)*

Section	Specification	MX Series	EX9200
	OFPFW_DL_SRC	Supported	Supported
	OFPFW_DL_DST	Supported	Supported
	OFPFW_DL_TYPE	Supported	Supported
	OFPFW_NW_PROTO	Supported	Supported
	OFPFW_TP_SRC	Supported	Supported
	OFPFW_TP_DST	Supported	Supported
	No wildcards set. Match entire 12 tuple.	Supported	Supported

Table 13 on page 45 lists the support for OpenFlow v1.0 flow actions.

Table 13: Junos OS Support for OpenFlow v1.0 Flow Actions

Section	Specification	MX Series	EX9200
5.2.4	OFPAT_OUTPUT:		
	OFPP_IN_PORT	Not supported	Not supported
	OFPP_TABLE	Not supported	Not supported
	OFPP_NORMAL	Supported	Supported
	OFPP_FLOOD	Supported	Supported
	OFPP_ALL	Supported	Supported
	OFPP_CONTROLLER	Supported	Supported
	OFPP_LOCAL	Not supported	Not supported
	OFPAT_SET_VLAN_VID	Supported	Supported
	OFPAT_SET_VLAN_PCP	Not supported	Not supported
	OFPAT_STRIP_VLAN	Supported	Supported
	OFPAT_SET_DL_SRC	Not supported	Not supported
	OFPAT_SET_DL_DST	Supported	Not supported
	OFPAT_SET_NW_SRC	Not supported	Not supported

Table 13: Junos OS Support for OpenFlow v1.0 Flow Actions *(continued)*

Section	Specification	MX Series	EX9200
	OFPAT_SET_NW_DST	Not supported	Not supported
	OFPAT_SET_NW_TOS	Not supported	Not supported
	OFPAT_SET_TP_SRC	Not supported	Not supported
	OFPAT_SET_TP_DST	Not supported	Not supported
	OFPAT_ENQUEUE	Not supported	Not supported

Table 14 on page 46 lists the support for OpenFlow v1.0 flow actions in Send Packet messages (OFPT_PACKET_OUT).

Table 14: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT)

Section	Specification	MX Series	EX9200
5.2.4	OFPAT_OUTPUT:		
	OFPP_IN_PORT	Not supported	Not supported
	OFPP_TABLE	Not supported	Not supported
	OFPP_NORMAL	Not supported	Not supported
	OFPP_FLOOD	Supported	Supported
	OFPP_ALL	Supported	Supported
	OFPP_CONTROLLER	Not supported	Not supported
	OFPP_LOCAL	Not supported	Not supported
	OFPAT_SET_VLAN_VID	Not supported	Not supported
	OFPAT_SET_VLAN_PCP	Not supported	Not supported
	OFPAT_STRIP_VLAN	Not supported	Not supported
	OFPAT_SET_DL_SRC	Not supported	Not supported
	OFPAT_SET_DL_DST	Not supported	Not supported
	OFPAT_SET_NW_SRC	Not supported	Not supported
	OFPAT_SET_NW_DST	Not supported	Not supported

Table 14: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT) *(continued)*

Section	Specification	MX Series	EX9200
	OFPAT_SET_NW_TOS	Not supported	Not supported
	OFPAT_SET_TP_SRC	Not supported	Not supported
	OFPAT_SET_TP_DST	Not supported	Not supported
	OFPAT_ENQUEUE	Not supported	Not supported

Table 15 on page 47 lists the support for OpenFlow v1.0 statistics.

Table 15: Junos OS Support for OpenFlow v1.0 Statistics

Section	Specification	MX Series	EX9200
5.3.5	OFPST_DESC	Supported	Supported
	OFPST_FLOW	Supported	Supported
	OFPST_AGGREGATE	Supported	Supported
	OFPST_TABLE	Supported	Supported
	OFPST_PORT	Supported	Supported
	OFPST_QUEUE	Supported	Supported
	OFPST_VENDOR	Gracefully ignored	Gracefully ignored

Table 16 on page 47 lists the support for OpenFlow v1.0 features.

Table 16: Junos OS Support for OpenFlow v1.0 Features

Section	Specification	MX Series	EX9200
4.4	Encryption. Controller and switch communicate through a TLS connection	Not supported	Not supported
5.3.3	Flow Idle Timeout	Supported	Supported
	Flow Hard Timeout	Supported	Supported

Table 16: Junos OS Support for OpenFlow v1.0 Features *(continued)*

Section	Specification	MX Series	EX9200
	Flow Priority	Supported	Supported

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20

OpenFlow Operational Mode Commands | 196

OpenFlow v1.0 Compliance Matrix for QFX5100 and EX4600 Switches

Table 9 on page 40 through Table 16 on page 47 list the OpenFlow v1.0 support for QFX5100 switches.

- Table 9 on page 40 lists support for message types.
- Table 10 on page 42 lists support for port structure flags.
- Table 11 on page 44 lists support for match conditions.
- Table 12 on page 44 lists support for wildcards.
- Table 13 on page 45 lists support for flow actions.
- Table 14 on page 46 lists support for flow actions in Send Packet messages (OFPT_PACKET_OUT).
- Table 15 on page 47 lists support for statistics.
- Table 16 on page 47 lists support for features.

Table 9 on page 40 lists the OpenFlow v1.0 message type support.

Table 17: Junos OS Support for OpenFlow v1.0 Message Types

Section	Specification	QFX5100 and EX4600
5.1	OFPT_HELLO	Supported
	OFPT_ERROR	Supported

Table 17: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPT_ECHO_REQUEST	Supported
	OFPT_ECHO_REPLY	Supported
	OFPT_VENDOR	Not supported
	OFPT_FEATURES_REQUEST	Supported
	OFPT_FEATURES_REPLY: Datapath ID N_buffers N_tables OFPC_FLOW_STATS OFPC_TABLE_STATS OFPC_PORT_STATS OFPC_STP OFPC_IP_REASM OFPC_QUEUE_STATS OFPC_ARP_MATCH_IP	Supported Supported -1 1 Supported Supported Supported Not supported Not supported Supported Not supported
	OFPT_GET_CONFIG_REQUEST	Supported
	OFPT_GET_CONFIG_REPLY	Supported
	OFPT_SET_CONFIG	Supported
	OFPT_PACKET_IN OFPT_PACKET_IN with buffer_id	Supported Not supported
	OFPT_FLOW_REMOVED	Supported
	OFPT_PORT_STATUS	Supported
	OFPT_PACKET_OUT OFPT_PACKET_OUT with buffer_id	Supported Not supported

Table 17: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPT_FLOW_MOD: OFPFC_ADD OFPFC_ADD with OFPFF_CHECK_OVERLAP OFPFC_MODIFY OFPFC_MODIFY_STRICT OFPFC_DELETE OFPFC_DELETE_STRICT OFPT_FLOW_MOD with buffer_id	Supported Supported Supported Supported Supported Supported Supported Not supported
	OFPT_PORT_MOD	Not supported
	OFPT_STATS_REQUEST	Supported
	OFPT_STATS_REPLY See Table 15 on page 47	Supported
	OFPT_BARRIER_REQUEST	Supported
	OFPT_BARRIER_REPLY	Supported
	OFPT_QUEUE_GET_CONFIG_REQUEST	Not supported
	OFPT_QUEUE_GET_CONFIG_REPLY	Not supported

Table 10 on page 42 lists the OpenFlow v1.0 port structure flag support

Table 18: Junos OS Support for OpenFlow v1.0 Port Structure Flags

Section	Specification	QFX5100 and EX4600
5.2.1	OFPPC_PORT_DOWN	Not supported
	OFPPC_NO_STP	Not supported
	OFPPC_NO_RECV	Not supported
	OFPPC_NO_RECV_STP	Not supported
	OFPPC_NO_FLOOD	Not supported

Table 18: Junos OS Support for OpenFlow v1.0 Port Structure Flags *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPPC_NO_FWD	Not supported
	OFPPC_NO_PACKET_IN	Not supported
	OFPPS_LINK_DOWN	Supported
	OFPPS_STP_LISTEN	Not supported
	OFPPS_STP_LEARN	Not supported
	OFPPS_STP_FORWARD	Not supported
	OFPPS_STP_BLOCK	Not supported
	OFPPS_STP_MASK	Not supported
	OFPPF_10MB_HD	Supported
	OFPPF_10MB_FD	Supported
	OFPPF_100MB_HD	Supported
	OFPPF_100MB_FD	Supported
	OFPPF_1GB_HD	Supported
	OFPPF_1GB_FD	Supported
	OFPPF_10GB_FD	Supported
	OFPPF_COPPER	Supported
	OFPPF_FIBER	Supported
	OFPPF_AUTONEG	Supported
	OFPPF_PAUSE	Not supported
	OFPPF_PAUSE_ASYM	Not supported

Table 11 on page 44 lists OpenFlow v1.0 match condition support.

Table 19: Junos OS Support for OpenFlow v1.0 Match Conditions

Section	Specification	QFX5100 and EX4600
5.2.3	dl_src (Ethernet source address)	Supported
	dl_dst (Ethernet destination address)	Supported
	dl_vlan (Input VLAN ID) **NOTE:** The flow match condition for the VLAN ID must be less than 4096. Otherwise, the flow is not installed. The only exception is VLAN ID 65535, which corresponds to untagged frames.	Supported
	dl_vlan_pcp (Input VLAN priority) **NOTE:** The flow match condition for the VLAN priority must be in accordance with 802.1p specifications. Otherwise, the flow is not installed.	Supported
	dl_type (Ethernet frame type)	Supported
	nw_tos (IP TOS (6-bit DSCP))	Supported
	nw_proto (IP Protocol or lower 8 bits of ARP opcode)	Supported
	nw_src (IP source address)	Supported
	nw_dst (IP destination address)	Supported
	tp_src (TCP/UDP source port/ICMPv4 type)	Supported
	tp_dst (TCP/UDP destination port/ICMPv4 code)	Supported
	Match all 12 tuples or a combination of tuples	Supported

Table 12 on page 44 lists the OpenFlow v1.0 wildcard support.

Table 20: Junos OS Support for OpenFlow v1.0 Wildcards

Section	Specification	QFX5100 and EX4600
5.2.3	OFPFW_IN_PORT	Supported
	OFPFW_DL_VLAN	Supported

Table 20: Junos OS Support for OpenFlow v1.0 Wildcards *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPFW_DL_SRC	Supported
	OFPFW_DL_DST	Supported
	OFPFW_DL_TYPE	Supported
	OFPFW_NW_PROTO	Supported
	OFPFW_TP_SRC	Supported
	OFPFW_TP_DST	Supported
	No wild cards set. Match entire 12 tuple.	Supported

Table 13 on page 45 lists the OpenFlow v1.0 flow action support.

Table 21: Junos OS Support for OpenFlow v1.0 Flow Actions

Section	Specification	QFX5100 and EX4600
5.2.4	OFPAT_OUTPUT:	
	OFPP_IN_PORT	Not supported
	OFPP_TABLE	Not supported
	OFPP_NORMAL	Not supported
	OFPP_FLOOD	Supported
	OFPP_ALL	Supported
	OFPP_CONTROLLER	Supported
	OFPP_LOCAL	Not supported
	OFPAT_SET_VLAN_VID	Supported
	OFPAT_SET_VLAN_PCP	Not supported
	OFPAT_STRIP_VLAN	Supported
	OFPAT_SET_DL_SRC	Not supported
	OFPAT_SET_DL_DST	Not supported
	OFPAT_SET_NW_SRC	Not supported

Table 21: Junos OS Support for OpenFlow v1.0 Flow Actions *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPAT_SET_NW_DST	Not supported
	OFPAT_SET_NW_TOS	Not supported
	OFPAT_SET_TP_SRC	Not supported
	OFPAT_SET_TP_DST	Not supported
	OFPAT_ENQUEUE	Not supported

Table 14 on page 46 lists the OpenFlow v1.0 flow action support in Send Packet messages (OFPT_PACKET_OUT).

Table 22: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT)

Section	Specification	QFX5100 and EX4600
5.2.4	OFPAT_OUTPUT:	
	OFPP_IN_PORT	Not supported
	OFPP_TABLE	Not supported
	OFPP_NORMAL	Not supported
	OFPP_FLOOD	Supported
	OFPP_ALL	Supported
	OFPP_CONTROLLER	Not supported
	OFPP_LOCAL	Not supported
	OFPAT_SET_VLAN_VID	Supported
	OFPAT_SET_VLAN_PCP	Not supported
	OFPAT_STRIP_VLAN	Supported
	OFPAT_SET_DL_SRC	Not supported
	OFPAT_SET_DL_DST	Not supported
	OFPAT_SET_NW_SRC	Not supported
	OFPAT_SET_NW_DST	Not supported

Table 22: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT) *(continued)*

Section	Specification	QFX5100 and EX4600
	OFPAT_SET_NW_TOS	Not supported
	OFPAT_SET_TP_SRC	Not supported
	OFPAT_SET_TP_DST	Not supported
	OFPAT_ENQUEUE	Not supported

Table 15 on page 47 lists the OpenFlow v1.0 statistics support.

Table 23: Junos OS Support for OpenFlow v1.0 Statistics

Section	Specification	QFX5100 and EX4600
5.3.5	OFPST_DESC	Supported
	OFPST_FLOW	Supported
	OFPST_AGGREGATE	Supported
	OFPST_TABLE	Supported
	OFPST_PORT	Supported
	OFPST_QUEUE	Not supported
	OFPST_VENDOR	Gracefully ignored

Table 16 on page 47 lists the OpenFlow v1.0 feature support.

Table 24: Junos OS Support for OpenFlow v1.0 Features

Section	Specification	QFX5100 and EX4600
4.4	Encryption. Controller and switch communicate through a TLS connection.	Not supported
5.3.3	Flow Idle Timeout	Supported
	Flow Hard Timeout	Supported

Table 24: Junos OS Support for OpenFlow v1.0 Features *(continued)*

Section	Specification	QFX5100 and EX4600
	Flow Priority	Supported

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | **20**
- OpenFlow Operational Mode Commands | **196**

OpenFlow v1.0 Compliance Matrix for EX4550 Switches

Table 9 on page 40 through Table 16 on page 47 list the OpenFlow v1.0 support for the EX4550 switch.

- Table 9 on page 40 lists support for message types.
- Table 10 on page 42 lists support for port structure flags.
- Table 11 on page 44 lists match condition support.
- Table 12 on page 44 lists wildcard support.
- Table 13 on page 45 lists flow action support.
- Table 14 on page 46 lists flow action in Send Packet messages (OFPT_PACKET_OUT) support.
- Table 15 on page 47 lists statistics support.
- Table 16 on page 47 lists feature support.

Table 9 on page 40 lists the OpenFlow v1.0 message type support.

Table 25: Junos OS Support for OpenFlow v1.0 Message Types

Section	Specification	EX4550
5.1	OFPT_HELLO	Supported
	OFPT_ERROR	Supported

Table 25: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	EX4550
	OFPT_ECHO_REQUEST	Supported
	OFPT_ECHO_REPLY	Supported
	OFPT_VENDOR	Not supported
	OFPT_FEATURES_REQUEST	Supported
	OFPT_FEATURES_REPLY: Datapath ID N_buffers N_tables OFPC_FLOW_STATS OFPC_TABLE_STATS OFPC_PORT_STATS OFPC_STP OFPC_IP_REASM OFPC_QUEUE_STATS OFPC_ARP_MATCH_IP	Supported Supported -1 1 Not supported Supported Supported Not supported Not supported Supported Not supported
	OFPT_GET_CONFIG_REQUEST	Supported
	OFPT_GET_CONFIG_REPLY	Supported
	OFPT_SET_CONFIG	Supported
	OFPT_PACKET_IN OFPT_PACKET_IN with buffer_id	Supported Not supported
	OFPT_FLOW_REMOVED	Supported
	OFPT_PORT_STATUS	Supported
	OFPT_PACKET_OUT OFPT_PACKET_OUT with buffer_id	Supported Not supported

Table 25: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	EX4550
	OFPT_FLOW_MOD: OFPFC_ADD OFPFC_ADD with OFPFF_CHECK_OVERLAP OFPFC_MODIFY OFPFC_MODIFY_STRICT OFPFC_DELETE OFPFC_DELETE_STRICT OFPT_FLOW_MOD with buffer_id	Supported Supported Supported Supported Supported Supported Supported Not supported
	OFPT_PORT_MOD	Not supported
	OFPT_STATS_REQUEST	Supported
	OFPT_STATS_REPLY See Table 15 on page 47	Supported
	OFPT_BARRIER_REQUEST	Supported
	OFPT_BARRIER_REPLY	Supported
	OFPT_QUEUE_GET_CONFIG_REQUEST	Not supported
	OFPT_QUEUE_GET_CONFIG_REPLY	Not supported

Table 10 on page 42 lists the OpenFlow v1.0 port structure flag support.

Table 26: Junos OS Support for OpenFlow v1.0 Port Structure Flags

Section	Specification	EX4550
5.2.1	OFPPC_PORT_DOWN	Not supported
	OFPPC_NO_STP	Not supported
	OFPPC_NO_RECV	Not supported
	OFPPC_NO_RECV_STP	Not supported
	OFPPC_NO_FLOOD	Not supported

Table 26: Junos OS Support for OpenFlow v1.0 Port Structure Flags *(continued)*

Section	Specification	EX4550
	OFPPC_NO_FWD	Not supported
	OFPPC_NO_PACKET_IN	Not supported
	OFPPS_LINK_DOWN	Supported
	OFPPS_STP_LISTEN	Not supported
	OFPPS_STP_LEARN	Not supported
	OFPPS_STP_FORWARD	Not supported
	OFPPS_STP_BLOCK	Not supported
	OFPPS_STP_MASK	Not supported
	OFPPF_10MB_HD	Supported
	OFPPF_10MB_FD	Supported
	OFPPF_100MB_HD	Supported
	OFPPF_100MB_FD	Supported
	OFPPF_1GB_HD	Supported
	OFPPF_1GB_FD	Supported
	OFPPF_10GB_FD	Supported
	OFPPF_COPPER	Supported
	OFPPF_FIBER	Supported
	OFPPF_AUTONEG	Supported
	OFPPF_PAUSE	Not supported
	OFPPF_PAUSE_ASYM	Not supported

Table 11 on page 44 lists OpenFlow v1.0 match condition support.

Table 27: Junos OS Support for OpenFlow v1.0 Match Conditions

Section	Specification	EX4550
5.2.3	dl_src (Ethernet source address)	Supported
	dl_dst (Ethernet destination address)	Supported
	dl_vlan (Input VLAN ID) **NOTE:** The flow match condition for the VLAN ID must be less than 4096. Otherwise, the flow is not installed. The only exception is VLAN ID 65535, which corresponds to untagged frames.	Supported
	dl_vlan_pcp (Input VLAN priority) **NOTE:** The flow match condition for the VLAN priority must be in accordance with 802.1p. Otherwise, the flow is not installed.	Supported
	dl_type (Ethernet frame type)	Supported
	nw_tos (IP TOS (6 bits DSCP))	Supported
	nw_proto (IP Protocol or lower 8 bits of ARP opcode)	Supported
	nw_src (IP source address)	Supported
	nw_dst (IP destination address)	Supported
	tp_src (TCP/UDP source port)	Supported
	tp_dst (TCP/UDP destination port)	Supported
	Match all 12 tuples or a combination of tuples	Supported
	OFPXMT_OFB_IN_PORT	Not supported
	OFPXMT_OFB_IN_PHY_PORT	Not supported
	OFPXMT_OFB_METADATA	Not supported
	OFPXMT_OFB_ETH_DST	Not supported
	OFPXMT_OFB_ETH_SRC	Not supported
	OFPXMT_OFB_ETH_TYPE	Not supported

Table 27: Junos OS Support for OpenFlow v1.0 Match Conditions *(continued)*

Section	Specification	EX4550
	OFPXMT_OFB_VLAN_VID	Not supported
	OFPXMT_OFB_VLAN_PCP	Not supported
	OFPXMT_OFB_IP_DSCP	Not supported
	OFPXMT_OFB_IP_ECN	Not supported
	OFPXMT_OFB_IP_PROTO	Not supported
	OFPXMT_OFB_IPV4_SRC	Not supported
	OFPXMT_OFB_IPV4_DST	Not supported
	OFPXMT_OFB_TCP_SRC	Not supported
	OFPXMT_OFB_TCP_DST	Not supported
	OFPXMT_OFB_UDP_SRC	Not supported
	OFPXMT_OFB_UDP_DST	Not supported
	OFPXMT_OFB_SCTP_SRC	Not supported
	OFPXMT_OFB_SCTP_DST	Not supported
	OFPXMT_OFB_ICMPV4_TYPE	Not supported
	OFPXMT_OFB_ICMPV4_CODE	Not supported
	OFPXMT_OFB_ARP_OP	Not supported
	OFPXMT_OFB_ARP_SPA	Not supported
	OFPXMT_OFB_ARP_TPA	Not supported
	OFPXMT_OFB_ARP_SHA	Not supported
	OFPXMT_OFB_ARP_THA	Not supported
	OFPXMT_OFB_IPV6_SRC	Not supported

Table 27: Junos OS Support for OpenFlow v1.0 Match Conditions *(continued)*

Section	Specification	EX4550
	OFPXMT_OFB_IPV6_DST	Not supported
	OFPXMT_OFB_IPV6_FLABEL	Not supported
	OFPXMT_OFB_ICMPV6_TYPE	Not supported
	OFPXMT_OFB_ICMPV6_CODE	Not supported
	OFPXMT_OFB_IPV6_ND_TARGET	Not supported
	OFPXMT_OFB_IPV6_ND_SLL	Not supported
	OFPXMT_OFB_IPV6_ND_TLL	Not supported
	OFPXMT_OFB_MPLS_LABEL	Not supported
	OFPXMT_OFB_MPLS_TC	Not supported
	OFPXMT_OFB_MPLS_BOS	Not supported
	OFPXMT_OFB_PBB_ISID	Not supported
	OFPXMT_OFB_TUNNEL_ID	Not supported
	OFPXMT_OFB_IPV6_EXTHDR	Not supported

Table 12 on page 44 lists the OpenFlow v1.0 wildcard support.

Table 28: Junos OS Support for OpenFlow v1.0 Wildcards

Section	Specification	EX4550
5.2.3	OFPFW_IN_PORT	Supported
	OFPFW_DL_VLAN	Supported
	OFPFW_DL_SRC	Supported
	OFPFW_DL_DST	Supported
	OFPFW_DL_TYPE	Supported

Table 28: Junos OS Support for OpenFlow v1.0 Wildcards *(continued)*

Section	Specification	EX4550
	OFPFW_NW_PROTO	Supported
	OFPFW_TP_SRC	Supported
	OFPFW_TP_DST	Supported
	No wild cards set. Match entire 12 tuple.	Supported

Table 13 on page 45 lists the OpenFlow v1.0 flow action support.

Table 29: Junos OS Support for OpenFlow v1.0 Flow Actions

Section	Specification	EX4550
5.2.4	OFPAT_OUTPUT:	
	OFPP_IN_PORT	Not supported
	OFPP_TABLE	Not supported
	OFPP_NORMAL	Supported
	OFPP_FLOOD	Supported
	OFPP_ALL	Supported
	OFPP_CONTROLLER	Supported
	OFPP_LOCAL	Not supported
	OFPAT_SET_VLAN_VID	Not supported
	OFPAT_SET_VLAN_PCP	Not supported
	OFPAT_STRIP_VLAN	Not supported
	OFPAT_SET_DL_SRC	Not supported
	OFPAT_SET_DL_DST	Not supported
	OFPAT_SET_NW_SRC	Not supported
	OFPAT_SET_NW_DST	Not supported
	OFPAT_SET_NW_TOS	Not supported
	OFPAT_SET_TP_SRC	Not supported

Table 29: Junos OS Support for OpenFlow v1.0 Flow Actions *(continued)*

Section	Specification	EX4550
	OFPAT_SET_TP_DST	Not supported
	OFPAT_ENQUEUE	Not supported

Table 14 on page 46 lists the OpenFlow v1.0 flow action support in Send Packet messages (OFPT_PACKET_OUT).

Table 30: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT)

Section	Specification	EX4550
5.2.4	OFPAT_OUTPUT:	
	OFPP_IN_PORT	Not supported
	OFPP_TABLE	Not supported
	OFPP_NORMAL	Not supported
	OFPP_FLOOD	Supported
	OFPP_ALL	Supported
	OFPP_CONTROLLER	Not supported
	OFPP_LOCAL	Not supported
	OFPAT_SET_VLAN_VID	Not supported
	OFPAT_SET_VLAN_PCP	Not supported
	OFPAT_STRIP_VLAN	Not supported
	OFPAT_SET_DL_SRC	Not supported
	OFPAT_SET_DL_DST	Not supported
	OFPAT_SET_NW_SRC	Not supported
	OFPAT_SET_NW_DST	Not supported
	OFPAT_SET_NW_TOS	Not supported
	OFPAT_SET_TP_SRC	Not supported
	OFPAT_SET_TP_DST	Not supported

Table 30: Junos OS Support for OpenFlow v1.0 Flow Actions in Send Packet Messages (OFPT_PACKET_OUT) *(continued)*

Section	Specification	EX4550
	OFPAT_ENQUEUE	Not supported

Table 15 on page 47 lists the OpenFlow v1.0 statistics support.

Table 31: Junos OS Support for OpenFlow v1.0 Statistics

Section	Specification	EX4550
5.3.5	OFPST_DESC	Supported
	OFPST_FLOW	Not supported
	OFPST_AGGREGATE	Not supported
	OFPST_TABLE	Supported
	OFPST_PORT	Supported
	OFPST_QUEUE	Supported
	OFPST_VENDOR	Gracefully ignored

Table 16 on page 47 lists the OpenFlow v1.0 feature support.

Table 32: Junos OS Support for OpenFlow v1.0 Features

Section	Specification	EX4550
4.4	Encryption. Controller and switch communicate through a TLS connection.	Not supported
5.3.3	Flow Idle Timeout	Not supported
	Flow Hard Timeout	Supported
	Flow Priority	Supported

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18
Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | 20
OpenFlow Operational Mode Commands | 196

OpenFlow v1.3.1 Compliance Matrix for Devices Running Junos OS

Starting with Junos OS Release 14.2R1, OpenFlow v1.3.1 support is introduced. The following tables list the support for OpenFlow v1.3.1 features on the indicated platforms.

- Table 33 on page 66 lists support for message types.
- Table 34 on page 69 lists support for features reply messages.
- Table 10 on page 42 lists support for port structure flags.
- Table 36 on page 70 lists support for port numbering.
- Table 11 on page 44 lists support for match conditions.
- Table 13 on page 45 lists support for flow actions.
- Table 39 on page 74 lists support for multipart messages.
- Table 40 on page 75 lists support for flow instructions.
- Table 41 on page 76 lists support for group types.

Table 33 on page 66 lists the support for OpenFlow v1.3.1 message types.

Table 33: Junos OS Support for OpenFlow v1.3.1 Message Types

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPT_HELLO	Supported	Supported	Supported
OFPT_ERROR	Supported	Supported	Supported
OFPT_ECHO_REQUEST	Supported	Supported	Supported
OFPT_ECHO_REPLY	Supported	Supported	Supported
OFPT_EXPERIMENTER	Not supported	Not supported	Not supported

Table 33: Junos OS Support for OpenFlow v1.3.1 Message Types *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPT_FEATURES_REQUEST	Supported	Supported	Supported
OFPT_FEATURES_REPLY See Table 34 on page 69.	Supported	Supported	Supported
OFPT_GET_CONFIG_REQUEST	Supported	Supported	Supported
OFPT_GET_CONFIG_REPLY	Supported	Supported	Supported
OFPT_SET_CONFIG	Supported	Supported	Supported
OFPT_PACKET_IN OFPT_PACKET_IN with buffer_id	Supported Not supported	Supported Not supported	Supported Not supported
OFPT_FLOW_REMOVED	Supported	Supported	Supported
OFPT_PORT_STATUS	Supported	Supported	Supported
OFPT_PACKET_OUT OFPT_PACKET_OUT with buffer_id	Supported Not supported	Supported Not supported	Supported Not supported
OFPT_FLOW_MOD OFPT_FLOW_MOD with buffer_id OFPFC_ADD OFPFC_ADD with OFPFF_CHECK_OVERLAP OFPFC_MODIFY OFPFC_MODIFY_STRICT OFPFC_DELETE OFPFC_DELETE_STRICT	Supported Not supported Supported Supported Supported Supported Supported Supported	Supported Not supported Supported Supported Supported Supported Supported Supported	Supported Not supported Supported Supported Supported Supported Supported Supported
Flow Modification Flags: OFPFF_SEND_FLOW_REM OFPFF_CHECK_OVERLAP OFPFF_RESET_COUNTS OFPFF_NO_PKT_COUNTS OFPFF_NO_BYT_COUNTS	Supported Supported Supported Supported Supported Supported	Supported Supported Supported Supported Supported Supported	Supported Supported Supported Supported Supported Supported

Table 33: Junos OS Support for OpenFlow v1.3.1 Message Types *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPT_GROUP_MOD: OFPGC_ADD OFPGC_MODIFY OFPGC_DELETE	Supported Supported Supported Supported	Supported Supported Supported Supported	Supported Supported Supported Supported
OFPT_PORT_MOD	Not supported	Not supported	Not supported
OFPT_TABLE_MOD	Not supported	Not supported	Not supported
OFPT_MULTIPART_REQUEST See Table 39 on page 74	Supported	Supported	Supported
OFPT_MULTIPART_REPLY See Table 39 on page 74	Supported	Supported	Supported
OFPT_BARRIER_REQUEST	Supported	Supported	Supported
OFPT_BARRIER_REPLY	Supported	Supported	Supported
OFPT_QUEUE_GET_CONFIG_REQUEST	Not supported	Not supported	Not supported
OFPT_QUEUE_GET_CONFIG_REPLY	Not supported	Not supported	Not supported
OFPT_ROLE_REQUEST	Not supported	Not supported	Not supported
OFPT_ROLE_REPLY	Not supported	Not supported	Not supported
OFPT_GET_ASYNC_REQUEST	Not supported	Not supported	Not supported
OFPT_GET_ASYNC_REPLY	Not supported	Not supported	Not supported
OFPT_SET_ASYNC	Not supported	Not supported	Not supported
OFPT_METER_MOD	Not supported	Not supported	Not supported
OFPT_VENDOR	Not supported	Not supported	Not supported

Table 34 on page 69 lists the support for OpenFlow v1.3.1 features reply messages.

Table 34: Junos OS Support for OpenFlow v1.3.1 Features Reply Messages

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPT_FEATURES_REPLY:			
Datapath ID	Supported	Supported	Supported
N_buffers	0	0	-1
N_tables	1	1	1
Auxiliary ID	0	0	0
OFPC_FLOW_STATS	Supported	Supported	Supported
OFPC_TABLE_STATS	Supported	Supported	Supported
OFPC_PORT_STATS	Supported	Supported	Supported
OFPC_GROUP_STATS	Supported	Supported	Supported
OFPC_IP_REASM	Not supported	Not supported	Not supported
OFPC_QUEUE_STATS	Supported	Supported	Supported
OFPC_PORT_BLOCKED	Not supported	Not supported	Not supported

Table 10 on page 42 lists the support for OpenFlow v1.3.1 port structure flags.

Table 35: Junos OS Support for OpenFlow v1.3.1 Port Structure Flags

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPPC_PORT_DOWN	Not supported	Not supported	Not supported
OFPPC_NO_STP	Not supported	Not supported	Not supported
OFPPC_NO_RECV	Not supported	Not supported	Not supported
OFPPC_NO_RECV_STP	Not supported	Not supported	Not supported
OFPPC_NO_FLOOD	Not supported	Not supported	Not supported
OFPPC_NO_FWD	Not supported	Not supported	Not supported
OFPPC_NO_PACKET_IN	Not supported	Not supported	Not supported
OFPPS_LINK_DOWN	Supported	Supported	Supported
OFPPS_BLOCKED	Not supported	Not supported	Not supported
OFPPS_LIVE	Not supported	Not supported	Not supported
OFPPF_10MB_HD	Supported	Supported	Supported

Table 35: Junos OS Support for OpenFlow v1.3.1 Port Structure Flags *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPPF_10MB_FD	Supported	Supported	Supported
OFPPF_100MB_HD	Supported	Supported	Supported
OFPPF_100MB_FD	Supported	Supported	Supported
OFPPF_1GB_HD	Supported	Supported	Supported
OFPPF_1GB_FD	Supported	Supported	Supported
OFPPF_10GB_FD	Supported	Supported	Supported
OFPPF_40GB-FD	Supported	Supported	Supported
OFPPF_100GB-FD	Supported	Supported	Not supported
OFPPF_1TB-FD	Not supported	Not supported	Not supported
OFPPF_COPPER	Supported	Supported	Not supported
OFPPF_FIBER	Supported	Supported	Supported
OFPPF_AUTONEG	Supported	Supported	Supported
OFPPF_PAUSE	Not supported	Not supported	Not supported
OFPPF_PAUSE_ASYM	Not supported	Not supported	Not supported

Table 36 on page 70 lists the support for OpenFlow v1.3.1 port numbering.

Table 36: Junos OS Support for OpenFlow v1.3.1 Port Numbering

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPP_IN_PORT	Not supported	Not supported	Not supported
OFPP_TABLE	Not supported	Not supported	Not supported
OFPP_NORMAL	Supported	Supported	Not supported

Table 36: Junos OS Support for OpenFlow v1.3.1 Port Numbering *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPP_FLOOD (all except input and STP disabled port) (Flood and All are same)	Supported	Supported	Supported
OFPP_ALL (all except input)	Supported	Supported	Supported
OFPP_CONTROLLER	Supported	Supported	Supported
OFPP_LOCAL	Not supported	Not supported	Not supported

Table 11 on page 44 lists the support for OpenFlow v1.3.1 match conditions.

Table 37: Junos OS Support for OpenFlow v1.3.1 Match Conditions

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPXMT_OFB_IN_PORT	Supported	Supported	Supported
OFPXMT_OFB_IN_PHY_PORT	Not supported	Not supported	Not supported
OFPXMT_OFB_METADATA	Not supported	Not supported	Not supported
OFPXMT_OFB_ETH_SRC	Supported	Supported	Supported
OFPXMT_OFB_ETH_DST	Supported	Supported	Supported
OFPXMT_OFB_VLAN_VID	Supported	Supported **NOTE:** Native VLAN is not supported on the OpenFlow logical interface when multiple logical interfaces are configured on that interface.	Supported
OFPXMT_OFB_VLAN_PCP	Supported	Supported	Supported
OFPXMT_OFB_ETH_TYPE	Supported	Supported	Supported

Table 37: Junos OS Support for OpenFlow v1.3.1 Match Conditions *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPXMT_OFB_IP_DSCP	Supported	Supported	Supported
OFPXMT_OFB_IP_ECN	Not supported	Not supported	Not supported
OFPXMT_OFB_IP_PROTO	Supported	Supported	Supported
OFPXMT_OFB_IPV4_SRC	Supported	Supported	Supported
OFPXMT_OFB_IPV4_DST	Supported	Supported	Supported
OFPXMT_OFB_TCP_SRC	Supported	Supported	Supported
OFPXMT_OFB_TCP_DST	Supported	Supported	Supported
OFPXMT_OFB_UDP_SRC	Supported	Supported	Supported
OFPXMT_OFB_UDP_DST	Supported	Supported	Supported
OFPXMT_OFB_SCTP_SRC	Not supported	Not supported	Not supported
OFPXMT_OFB__SCTP_DST	Not supported	Not supported	Not supported
OFPXMT_OFB_ICMPV4_TYPE	Supported	Supported	Supported
OFPXMT_OFB_ICMPV4_CODE	Supported	Supported	Supported
OFPXMT_OFB_ARP_OP	Not supported	Not supported	Not supported
OFPXMT_OFB_ARP_SPA	Not supported	Not supported	Not supported
OFPXMT_OFB_ARP_TPA	Not supported	Not supported	Not supported
OFPXMT_OFB_ARP_SHA	Not supported	Not supported	Not supported
OFPXMT_OFB_ARP_THA	Not supported	Not supported	Not supported
OFPXMT_OFB_IPV6_SRC	Supported	Not supported	Not supported
OFPXMT_OFB_IPV6_DST	Supported	Not supported	Not supported

Table 37: Junos OS Support for OpenFlow v1.3.1 Match Conditions *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPXMT_OFB_IPV6_FLABEL	Not supported	Not supported	Not supported
OFPXMT_OFB_ICMPV6_TYPE	Not supported	Not supported	Not supported
OFPXMT_OFB_ICMPV6_CODE	Not supported	Not supported	Not supported
OXM_OF_IPV6_ND_TARGET	Not supported	Not supported	Not supported
OXM_OF_IPV6_ND_SLL	Not supported	Not supported	Not supported
OXM_OF_IPV6_ND_TLL	Not supported	Not supported	Not supported
OXM_OF_IPV6_EXTHDR	Not supported	Not supported	Not supported
OFPXMT_OFB_MPLS_LABEL	Not supported	Not supported	Not supported
OFPXMT_OFB_MPLS_TC	Not supported	Not supported	Not supported
OFPXMT_OFB_MPLS_BOS	Not supported	Not supported	Not supported
OFPXMT_OFB_PBB_ISID	Not supported	Not supported	Not supported
OFPXMT_OFB_TUNNEL_ID	Not supported	Not supported	Not supported

NOTE: The Junos OS implementation of OpenFlow v1.3.1 supports wildcards for all match conditions.

The Junos OS implementation of OpenFlow v1.3.1 does not support arbitrary bit masks for any fields or IPv6 addresses. This implementation supports only continuous masks for IPv4 and IPv6 source and destination addresses.

Table 13 on page 45 lists the support for OpenFlow v1.3.1 flow actions.

Table 38: Junos OS Support for OpenFlow v1.3.1 Flow Actions

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPAT_SET_VLAN_VID	Supported	Supported	Supported
OFPAT_SET_VLAN_PCP	Not supported	Not supported	Not supported
OFPAT_POP_VLAN	Supported	Supported	Supported
OFPAT_GROUP	Supported	Supported	Supported
OFPAT_COPY_TTL_OUT	Not supported	Not supported	Not supported
OFPAT_COPY_TTL_IN	Not supported	Not supported	Not supported
OFPAT_SET_MPLS_TTL	Not supported	Not supported	Not supported
OFPAT_DEC_MPLS_TTL	Not supported	Not supported	Not supported
OFPAT_PUSH_VLAN	Not supported	Not supported	Not supported
OFPAT_PUSH_MPLS	Not supported	Not supported	Not supported
OFPAT_POP_MPLS	Not supported	Not supported	Not supported
OFPAT_SET_QUEUE	Not supported	Not supported	Not supported
OFPAT_SET_NW_TTL	Not supported	Not supported	Not supported
OFPAT_DEC_NW_TTL	Not supported	Not supported	Not supported
OFPAT_PUSH_PBB	Not supported	Not supported	Not supported
OFPAT_POP_PBB	Not supported	Not supported	Not supported
OFPAT_EXPERIMENTER	Not supported	Not supported	Not supported

Table 39 on page 74 lists the support for OpenFlow v1.3.1 multipart messages.

Table 39: Junos OS Support for OpenFlow v1.3.1 Multipart Messages

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPMP_DESC	Supported	Supported	Supported

Table 39: Junos OS Support for OpenFlow v1.3.1 Multipart Messages *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPMP_FLOW	Supported	Supported	Supported
OFPMP_AGGREGATE	Supported	Supported	Supported
OFPMP_TABLE	Supported	Supported	Supported
OFPMP_PORT_STATS	Supported	Supported	Supported
OFPMP_QUEUE	Supported	Supported	Supported
OFPMP_GROUP	Supported	Supported	Supported
OFPMP_GROUP_DESC	Supported	Supported	Supported
OFPMP_GROUP_FEATURES	Supported	Supported	Supported
OFPMP_METER	Not supported	Not supported	Not supported
OFPMP_METER_CONFIG	Not supported	Not supported	Not supported
OFPMP_METER_FEATURES	Not supported	Not supported	Not supported
OFPMP_TABLE_FEATURES	Supported	Supported	Supported
OFPMP_PORT_DESC	Supported	Supported	Supported
OFPMP_EXPERIMENTER	Not supported	Not supported	Not supported

Table 40 on page 75 lists the support for OpenFlow v1.3.1 flow instructions.

NOTE: A flow can have a maximum of one of the supported flow instructions listed in Table 40 on page 75.

Table 40: Junos OS Support for OpenFlow v1.3.1 Flow Instructions

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPIT_GOTO_TABLE	Not supported	Not supported	Not supported

Table 40: Junos OS Support for OpenFlow v1.3.1 Flow Instructions *(continued)*

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPIT_WRITE_METADATA	Not supported	Not supported	Not supported
OFPIT_WRITE_ACTIONS	Supported	Supported	Supported
OFPIT_APPLY_ACTIONS	Supported	Supported	Supported
OFPIT_CLEAR_ACTIONS	Not supported	Not supported	Not supported
OFPIT_METER	Not supported	Not supported	Not supported
OFPIT_EXPERIMENTER	Not supported	Not supported	Not supported

Table 41 on page 76 lists the support for OpenFlow v1.3.1 group types.

Table 41: Junos OS Support for OpenFlow v1.3.1 Group Types

Specification	MX Series	EX9200	QFX5100 and EX4600
OFPGT_ALL	Supported	Supported	Supported
OFPGT_SELECT	Not supported	Not supported	Not supported
OFPGT_INDIRECT	Supported	Supported	Supported
OFPGT_FF	Not supported	Not supported	Not supported

Release History Table

Release	Description
14.2R1	Starting with Junos OS Release 14.2R1, OpenFlow v1.3.1 support is introduced.

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS | **20**
- Understanding How the OpenFlow Group Action Works | **33**
- OpenFlow Operational Mode Commands | **196**

2

CHAPTER

OpenFlow Basic Configuration

Configuring Support for OpenFlow on MX Series Routers | **78**

Example: Enabling OpenFlow on MX Series Routers | **81**

Configuring Support for OpenFlow on EX9200 Switches | **88**

Example: Enabling OpenFlow on EX9200 Switches | **92**

Configuring Support for OpenFlow on QFX5100 and EX4600 Switches | **99**

Example: Enabling OpenFlow on QFX5100 and EX4600 Switches | **102**

Configuring Support for OpenFlow on EX4550 Switches | **108**

Example: Enabling OpenFlow on EX4550 Switches | **111**

Configuring Support for OpenFlow on MX Series Routers

IN THIS SECTION

- Configuring the OpenFlow Interfaces | 78
- Configuring the OpenFlow Protocol | 79
- Configuring the OpenFlow Routing Instance | 80

The following sections configure MX Series routers to support OpenFlow using interfaces that participate solely in OpenFlow. For information about configuring hybrid interfaces, which concurrently support OpenFlow logical interfaces and non-OpenFlow logical interfaces, see "Configuring OpenFlow Hybrid Interfaces on MX Series Routers" on page 120.

Before configuring support for OpenFlow, ensure that the router meets the following requirements:

- MX Series router running Junos OS Release 13.3 or a later release
- OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed
- TCP connection between the router and an OpenFlow controller
- Connection between the management interface of the router and the management network, which is reachable from the controller IP address

Configuration tasks are described in detail in the following sections:

Configuring the OpenFlow Interfaces

You must configure interfaces participating in OpenFlow as Layer 2 interfaces. On MX Series routers, you configure the interfaces with encapsulation **ethernet-bridge** and protocol family **bridge**.

To configure the OpenFlow Interfaces:

- Configure the physical link-layer encapsulation type and the logical interface and protocol family.

  [edit interfaces *interface-name*]

```
user@host# set encapsulation ethernet-bridge
user@host# set unit unit family bridge
```

Configuring the OpenFlow Protocol

To configure support for OpenFlow, create a virtual switch instance, and specify a switch name, which must be 60 characters or less. For the virtual switch instance, configure the OpenFlow controller information and the participating logical interfaces. Optionally, configure the default action for packets that do not match a flow entry, the purge timer for invalid flows, and any OpenFlow traceoptions.

To configure the OpenFlow protocol:

1. Configure the OpenFlow controller IP address and the connection protocol.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set controller address address
   user@host# set controller protocol tcp
   ```

2. Specify the logical interfaces participating in OpenFlow under this virtual switch instance.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set interfaces interface-name1.unit1
   user@host# set interfaces interface-name2.unit1
   ```

3. (Optional) Configure the **default-action** statement for packets that do not match on an existing flow entry.

   If you do not configure the **default-action** statement, the default is **packet-in**, which indicates that packets with no matching flow entry must be sent to the controller for processing.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set default-action (drop | packet-in)
   ```

4. (Optional) Configure the **purge-flow-timer** statement, which is the number of seconds after which an invalid flow is purged from the flow table.

   ```
   [edit protocols openflow switch switch-name]
   ```

```
user@host# set purge-flow-timer seconds
```

5. (Optional) Configure OpenFlow traceoptions.

   If you do not configure a log filename, OpenFlow trace messages are logged in the default OpenFlow log file **/var/log/ofd** .

```
[edit protocols openflow]
user@host# set traceoptions flag flag
user@host# set traceoptions file file-name
```

Configuring the OpenFlow Routing Instance

To configure the virtual switch routing instance for OpenFlow traffic:

1. Configure the routing instance type as **virtual-switch**.

```
[edit routing-instances routing-instance-name]
user@host# set instance-type virtual-switch
```

2. Configure the bridge domain name and type.

```
[edit routing-instances routing-instance-name]
user@host# set bridge-domains name domain-type bridge
```

3. Configure the VLAN ID as **none**.

```
[edit routing-instances routing-instance-name]
user@host# set bridge-domains name vlan-id none
```

4. Configure the OpenFlow logical interfaces that will be bound to the routing instance.

```
[edit routing-instances routing-instance-name]
user@host# set bridge-domains name interface interface-name1.unit1
user@host# set bridge-domains name interface interface-name2.unit1
```

5. (Optional) If you use the NORMAL forward action to forward OpenFlow traffic using traditional Layer 2 and Layer 3 processing, configure an integrated routing and bridging (IRB) interface, and include the appropriate logical interface in the bridge domain configuration.

```
[edit routing-instances routing-instance-name]
user@host# set bridge-domains name routing-interface irb.unit
```

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Example: Enabling OpenFlow on MX Series Routers | 81

OpenFlow Operational Mode Commands | 196

openflow (Protocols OpenFlow) | 183

Example: Enabling OpenFlow on MX Series Routers

IN THIS SECTION

- Requirements | 81
- Overview | 82
- Configuration | 82
- Verification | 86

OpenFlow is an open standard that allows you to control traffic paths in a network by creating, deleting, and modifying flows in each device along a path. This example shows how to configure OpenFlow support on an MX240 router running Junos OS.

Requirements

This example uses the following hardware and software components:

- MX240 router running Junos OS Release 13.3 or a later release
- OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed
- TCP connection between the router and an OpenFlow controller
- Connection between the management interface of the router and the management network, which is reachable from the OpenFlow controller IP address

Overview

In this example, you configure support for OpenFlow on an MX240 router. The router has three interfaces that participate solely in OpenFlow: ge-1/0/0.0, ge-1/1/0.0, and xe-0/0/0.0. You first configure the interfaces as Layer 2 interfaces using physical link-layer encapsulation type **ethernet-bridge** and protocol family **bridge**.

MX Series routers require a separate virtual switch routing instance to isolate the OpenFlow traffic from the normal network traffic. This example configures a virtual switch routing instance, rt-bd-1, using instance type **virtual-switch** at the **[edit routing-instances]** hierarchy level. Within the routing instance, the bridge domain **of-bridge** includes all of the logical interfaces participating in OpenFlow.

You configure the OpenFlow virtual switch and OpenFlow protocol statements at the **[edit protocols openflow]** hierarchy level. In this example, the virtual switch, OFswitch1, connects to the controller over a TCP connection at IP address 172.16.1.1. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow, and OpenFlow traffic will only enter or exit from these interfaces.

Within the OpenFlow configuration, the **default-action** statement indicates the action the switch must take for packets that do not have a matching flow entry. If you omit the **default-action** statement, the default action is **packet-in**, which indicates that packets with no matching flow entry must be sent to the controller for processing. This example explicitly configures the default action for packets that do not have a matching flow entry as **packet-in**.

This example also configures OpenFlow traceoptions. In this case, the **flag all** statement indicates that all OpenFlow trace events should be captured and logged. Since the example does not configure a specific filename for the log file, OpenFlow trace messages are logged in the default OpenFlow log file **/var/log/ofd**.

Configuration

CLI Quick Configuration

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

```
set interfaces ge-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
set interfaces ge-1/1/0 encapsulation ethernet-bridge unit 0 family bridge
set interfaces xe-0/0/0 encapsulation ethernet-bridge unit 0 family bridge
set routing-instances rt-bd-1 instance-type virtual-switch
set routing-instances rt-bd-1 bridge-domains of-bridge vlan-id none
set routing-instances rt-bd-1 bridge-domains of-bridge interface ge-1/0/0.0
set routing-instances rt-bd-1 bridge-domains of-bridge interface ge-1/1/0.0
set routing-instances rt-bd-1 bridge-domains of-bridge interface xe-0/0/0.0
set protocols openflow switch OFswitch1 controller address 172.16.1.1
set protocols openflow switch OFswitch1 controller protocol tcp
set protocols openflow switch OFswitch1 interfaces ge-1/0/0.0
set protocols openflow switch OFswitch1 interfaces ge-1/1/0.0
set protocols openflow switch OFswitch1 interfaces xe-0/0/0.0
set protocols openflow switch OFswitch1 default-action packet-in
set protocols openflow traceoptions flag all
```

Step-by-Step Procedure

To configure support for OpenFlow:

1. Configure the OpenFlow interfaces as Layer 2 interfaces.

```
[edit interfaces]
user@host# set ge-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set ge-1/1/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set xe-0/0/0 encapsulation ethernet-bridge unit 0 family bridge
```

2. Configure the virtual switch routing instance.

```
[edit routing-instances]
user@host# set rt-bd-1 instance-type virtual-switch
user@host# set rt-bd-1 bridge-domains of-bridge vlan-id none
user@host# set rt-bd-1 bridge-domains of-bridge interface ge-1/0/0.0
user@host# set rt-bd-1 bridge-domains of-bridge interface ge-1/1/0.0
user@host# set rt-bd-1 bridge-domains of-bridge interface xe-0/0/0.0
```

3. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch OFswitch1]
user@host# set controller address 172.16.1.1
user@host# set controller protocol tcp
```

4. Configure the logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch OFswitch1]
user@host# set interfaces ge-1/0/0.0
user@host# set interfaces ge-1/1/0.0
user@host# set interfaces xe-0/0/0.0
```

5. Configure the default action for packets that do not have a matching flow entry.

```
[edit protocols openflow switch OFswitch1]
user@host# set default-action packet-in
```

6. Configure OpenFlow traceoptions.

```
[edit protocols openflow]
user@host# set traceoptions flag all
```

7. Commit the configuration.

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

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols openflow**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
ge-1/0/0 {
    encapsulation ethernet-bridge;
```

```
    unit 0 {
        family bridge;
    }
}
ge-1/1/0 {
    encapsulation ethernet-bridge;
    unit 0 {
        family bridge;
    }
}
xe-0/0/0 {
    encapsulation ethernet-bridge;
    unit 0 {
        family bridge;
    }
}
```

```
user@host# show protocols openflow
switch OFswitch1 {
    default-action packet-in;
    interfaces {
        ge-1/0/0.0;
        ge-1/1/0.0;
        xe-0/0/0.0;
    }
    controller {
        address 172.16.1.1;
        protocol tcp;
    }
}
traceoptions {
    flag all;
}
```

```
user@host# show routing-instances
rt-bd-1 {
    instance-type virtual-switch;
    bridge-domains {
        of-bridge {
            vlan-id none;
            interface ge-1/0/0.0;
            interface ge-1/1/0.0;
            interface xe-0/0/0.0;
```

```
    }
  }
}
```

Verification

IN THIS SECTION

- Verifying that the OpenFlow Controller Connection is Up | 86
- Verifying that the OpenFlow Interfaces Are Up | 87

Confirm that the configuration is working properly.

Verifying that the OpenFlow Controller Connection is Up

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command, and verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch should automatically initiate a connection to the controller after you commit the configuration.

```
user@host> show openflow controller
```

```
Openflowd controller information:
Controller socket: 11
Controller IP address: 172.16.1.1
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 1
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying that the OpenFlow Interfaces Are Up

Purpose

Verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

`user@host>` **show openflow interfaces**

```
Switch name: OFswitch1
Interface Name: ge-1/0/0.0
Interface port number: 41507
Interface Hardware Address: 00:00:5e:00:53:b1
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/0.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:b2
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: xe-0/0/0.0
Interface port number: 45549
Interface Hardware Address: 00:00:5e:00:53:b3
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Configuring Support for OpenFlow on MX Series Routers | **78**
- OpenFlow Operational Mode Commands | **196**
- openflow (Protocols OpenFlow) | **183**

Configuring Support for OpenFlow on EX9200 Switches

IN THIS SECTION

- Configuring the OpenFlow Interfaces | **89**
- Configuring the OpenFlow Protocol | **90**
- Configuring the OpenFlow Routing Instance | **91**

The following sections detail one method to configure EX9200 switches to support OpenFlow, using interfaces that participate solely in OpenFlow. For information about configuring hybrid interfaces, which concurrently support OpenFlow logical interfaces and non-OpenFlow logical interfaces, see “Configuring OpenFlow Hybrid Interfaces on EX9200 Switches” on page 134.

Before you begin configuring support for OpenFlow, ensure that the switch meets the following hardware and software requirements:

- EX9200 switch running Junos OS Release 13.3 or a later release.
- OpenFlow software package with a software package release that matches the Junos OS release running on the switch

- TCP connection between the switch and an OpenFlow controller
- Connection between the management interface of the switch and the management network, which is reachable from the controller IP address

Configuration tasks are described in detail in the following sections:

Configuring the OpenFlow Interfaces

To configure the OpenFlow interfaces:

1. Specify the VLAN tagging to be used and configure the encapsulation type.

   ```
   [edit interfaces interface-name]
   user@host# set flexible-vlan-tagging
   user@host# set encapsulation flexible-ethernet-services
   ```

2. Configure the logical interface and the protocol family.

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

3. Configure the interface as a trunk interface and specify the VLAN members associated with OpenFlow.

   ```
   [edit interfaces interface-name]
   user@host# set unit unit family ethernet-switching interface-mode trunk
   user@host# set unit unit family ethernet-switching vlan members openflow-vlan-ids
   ```

Configuring the OpenFlow Protocol

To configure support for OpenFlow, create a virtual switch instance, and specify a switch name, containing a maximum of 60 characters. For the virtual switch instance, configure the OpenFlow controller information and the participating logical interfaces. Optionally, configure the default action for packets that do not have a matching flow entry, the purge timer for invalid flows, and any OpenFlow traceoptions.

To configure the OpenFlow protocol:

1. Configure the OpenFlow controller IP address and TCP as the connection protocol.

```
[edit protocols openflow switch switch-name]
user@host# set controller address address
user@host# set controller protocol tcp
```

2. Specify the logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch switch-name]
user@host# set interfaces interface-name1.unit1
user@host# set interfaces interface-name2.unit1
```

3. (Optional) Configure the **default-action** statement for packets that do not match on an existing flow entry.

   If you do not configure the **default-action** statement, the default is **packet-in**, which indicates that packets that do not have a matching flow entry must be sent to the controller for processing.

```
[edit protocols openflow switch switch-name]
user@host# set default-action (drop | packet-in)
```

4. (Optional) Configure the **purge-flow-timer** statement, which is the number of seconds after which an invalid flow entry is purged from the flow table.

```
[edit protocols openflow switch switch-name]
user@host# set purge-flow-timer seconds
```

5. (Optional) Configure OpenFlow traceoptions.

   If you do not configure a log filename, OpenFlow trace messages are logged in the default OpenFlow log file **/var/log/ofd**.

   ```
   [edit protocols openflow]
   user@host# set traceoptions flag flag
   user@host# set traceoptions file file-name
   ```

Configuring the OpenFlow Routing Instance

To configure the virtual switch routing instance for OpenFlow traffic:

1. Configure the routing instance type as **virtual-switch**.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set instance-type virtual-switch
   ```

2. Configure the OpenFlow logical interfaces that will be bound to the routing instance.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set interface interface-name1.unit1
   user@host# set interface interface-name2.unit1
   ```

3. Configure the OpenFlow VLAN members under the **vlans** hierarchy.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set vlans name (vlan-id | vlan-id-list) openflow-vlan-ids
   ```

4. (Optional) If you use the NORMAL forward action to forward OpenFlow traffic using traditional Layer 2 and Layer 3 processing, configure an integrated routing and bridging (IRB) interface, and bind the appropriate logical interface to the VLAN.

   ```
   [edit routing-instances routing-instance-name]
   user@host# set vlans name l3-interface irb.unit
   ```

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

OpenFlow Operational Mode Commands | 196

openflow (Protocols OpenFlow) | 183

Example: Enabling OpenFlow on EX9200 Switches

IN THIS SECTION

- Requirements | 92
- Overview and Topology | 93
- Configuration | 93
- Verification | 97

OpenFlow is an open standard that enables you to control traffic paths in a network by creating, deleting, and modifying flows in each device, including EX9200 switches that have an OpenFlow software package installed, along a path. This example shows how to configure OpenFlow support on an EX9200 switch.

Requirements

This example uses the following hardware and software components:

- An EX9200 switch running Junos OS Release 13.3 or a later release.
- An OpenFlow software package is installed on the switch, and the software package release matches the Junos OS release running on the switch.
- The switch has a TCP connection to an OpenFlow controller, which needs to access the data plane of the switch.
- The switch is connected to the management network through the me0 interface and is reachable from the OpenFlow controller IP address.

Overview and Topology

In this example, you configure support for OpenFlow on an EX9200 switch. The switch has three interfaces that are dedicated to handling OpenFlow traffic: ge-1/0/0.0, ge-1/1/0.0, and xe-0/0/0.0.

EX9200 switches require a separate routing instance for a virtual switch. This routing instance isolates the experimental OpenFlow traffic from the normal network traffic. In this example, you configure a routing instance for the virtual switch, **OF-ri**, by using the instance type **virtual-switch** at the **[edit routing-instances]** hierarchy level. The routing instance **OF-ri** includes all of the logical interfaces participating in OpenFlow.

The virtual switch, **OFswitch1**, connects to the OpenFlow controller over a TCP connection at the IP address 198.51.100.174. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow, and the OpenFlow traffic only either enters or exits these interfaces.

A flow entry consists of a match condition against which packets entering an OpenFlow interface are compared, and the action that is applied to packets that match the condition. Each OpenFlow interface can have one or more flow entries. The **default-action** statement in the OpenFlow configuration indicates the action the switch applies for packets that do not have a matching flow entry. If you do not explicitly configure the **default-action** statement, the default action is **packet-in**, which indicates that packets that have no matching flow entry are sent to the OpenFlow controller for processing. In this example, you explicitly configure **packet-in** as the default action for packets that do not have a matching flow entry.

In this example, you configure OpenFlow traceoptions also. When traceoptions are configured with the **flag all** statement, all OpenFlow events are captured and logged. In this example, a specific filename is not configured for the log file. Therefore, OpenFlow events are logged in the default OpenFlow log file at **/var/log/ofd**.

Configuration

CLI Quick Configuration

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

```
set interfaces ge-1/0/0 unit 0 family ethernet-switching
set interfaces ge-1/1/0 unit 0 family ethernet-switching
set interfaces xe-0/0/0 unit 0 family ethernet-switching
set routing-instances OF-ri instance-type virtual-switch
set routing-instances OF-ri interface ge-1/0/0.0
set routing-instances OF-ri interface ge-1/1/0.0
set routing-instances OF-ri interface xe-0/0/0.0
```

```
set routing-instances OF-ri vlans of-bridge vlan-id none
set protocols openflow switch OFswitch1 controller address 198.51.100.174
set protocols openflow switch OFswitch1 controller protocol tcp port 6633
set protocols openflow switch OFswitch1 interfaces ge-1/0/0.0
set protocols openflow switch OFswitch1 interfaces ge-1/1/0.0
set protocols openflow switch OFswitch1 interfaces xe-0/0/0.0
set protocols openflow switch OFswitch1 default-action packet-in
set protocols openflow traceoptions flag all
```

Step-by-Step Procedure

To configure support for OpenFlow:

1. Configure the OpenFlow interfaces as Layer 2 interfaces:

```
[edit interfaces]
user@switch# set ge-1/0/0 unit 0 family ethernet-switching
user@switch# set ge-1/1/0 unit 0 family ethernet-switching
user@switch# set xe-0/0/0 unit 0 family ethernet-switching
```

2. Configure the virtual switch routing instance:

```
[edit routing-instances]
user@switch# set OF-ri instance-type virtual-switch
user@switch# set OF-ri interface ge-1/0/0.0
user@switch# set OF-ri interface ge-1/1/0.0
user@switch# set OF-ri interface xe-0/0/0.0
user@switch# set OF-ri vlans of-bridge vlan-id none
```

3. Configure the OpenFlow controller IP address and the connection protocol:

```
[edit protocols openflow switch OFswitch1]
user@switch# set controller address 198.51.100.174
user@switch# set controller protocol tcp port 6633
```

4. Configure the logical interfaces participating in OpenFlow under this virtual switch instance:

```
[edit protocols openflow switch OFswitch1]
user@switch# set interfaces ge-1/0/0.0
user@switch# set interfaces ge-1/1/0.0
```

```
user@switch# set interfaces xe-0/0/0.0
```

5. Configure the default action for packets that do not have a matching flow entry:

```
[edit protocols openflow switch OFswitch1]
user@switch# set default-action packet-in
```

6. Configure OpenFlow traceoptions:

```
[edit protocols openflow]
user@switch# set traceoptions flag all
```

7. Commit the configuration:

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

Results

From operational mode, display the results of your configuration by entering the **show configuration interfaces**, **show configuration protocols openflow**, and **show configuration routing-instances** commands. If the output does not display the specified configuration, repeat the instructions in this example to correct the configuration.

```
user@switch> show configuration interfaces
ge-1/0/0 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-1/1/0 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/0/0 {
    unit 0 {
        family ethernet-switching;
```

```
    }
}
```

```
user@switch> show configuration protocols openflow
switch OFswitch1 {
    default-action {
        packet-in;
    }
    interfaces {
        ge-1/0/0.0;
        ge-1/1/0.0;
        xe-0/0/0.0;
    }
    controller {
        address 198.51.100.174;
        protocol tcp {
            port 6633;
    }
}
traceoptions {
    flag all;
}
```

```
user@switch> show configuration routing-instances
OF-ri {
    instance-type virtual-switch;
    interface ge-1/0/0.0;
    interface ge-1/1/0.0;
    interface xe-0/0/0.0;
    vlans {
        of-bridge {
            vlan-id none;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the OpenFlow Controller Connection | 97
- Verifying the OpenFlow Interfaces | 97

Confirm that the configuration is working properly.

Verifying the OpenFlow Controller Connection

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command to verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch automatically initiates a connection to the controller after you commit the configuration.

```
user@switch> show openflow controller
```

```
Openflowd controller information:
Controller socket: 11
Controller IP address: 198.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 5
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying the OpenFlow Interfaces

Purpose

Verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

```
user@switch> show openflow interfaces
```

```
Switch name: OFswitch1
Interface Name: ge-1/0/0.0
Interface port number: 41507
Interface Hardware Address: 00:00:5E:00:53:b1
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/0.0
Interface port number: 44538
Interface Hardware Address: 00:00:5E:00:53:b2
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: xe-0/0/0.0
Interface port number: 45549
Interface Hardware Address: 00:00:5E:00:53:b3
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | **18**

Configuring Support for OpenFlow on EX9200 Switches | **88**

OpenFlow Operational Mode Commands | 196
openflow (Protocols OpenFlow) | 183

Configuring Support for OpenFlow on QFX5100 and EX4600 Switches

IN THIS SECTION

- Configuring the OpenFlow Interfaces | 100
- Configuring the OpenFlow Protocol | 100

This topic describes how to configure QFX5100 and EX4600 switches with interfaces that participate solely in OpenFlow.

Before configuring support for OpenFlow, ensure that the switch meets the following requirements:

- QFX5100 switch running Junos OS Release 14.1X53-D10 or later, or EX4600 switch running Junos OS Release 17.1R1 or later
- OpenFlow software package with a release that matches the Junos OS release running on the switch
- TCP connection between the switch and an OpenFlow controller
- Connection between the management interface (em0 or em1) of the switch and the management network

Configuration tasks are described in detail in the following sections:

Configuring the OpenFlow Interfaces

You must configure interfaces participating in OpenFlow as Layer 2 interfaces. On the switches, you configure the interfaces with protocol family **ethernet-switching**. Also, you can configure only a single logical port by specifying logical unit number 0.

To configure the OpenFlow interfaces:

- Configure the logical interface and protocol family.

```
[edit interfaces interface-name]
user@switch# set unit 0 family ethernet-switching
```

Configuring the OpenFlow Protocol

To configure support for OpenFlow, you must create a virtual switch, and then, configure the connection with the OpenFlow controller and the logical interfaces participating in OpenFlow for the virtual switch. Optionally, configure the default action for packets that do not match a flow entry, the purge timer for invalid flows, and any OpenFlow traceoptions.

To configure the OpenFlow protocol:

1. Create an OpenFlow virtual switch, and specify a switch name, which can contain a maximum of 60 characters.

```
[edit protocols openflow]
user@switch# set switch switch-name
```

2. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch switch-name]
user@switch# set controller address address
user@switch# set controller protocol tcp
```

3. Specify the logical interfaces participating in OpenFlow under this virtual switch.

```
[edit protocols openflow switch switch-name]
user@switch# set interfaces interface-name1.0
```

```
user@switch# set interfaces interface-name2.0
```

4. (Optional) Configure the **default-action** statement for packets that do not match an existing flow entry.

   If you do not configure the **default-action** statement, the default is **packet-in**, which indicates that packets with no matching flow entry are sent to the controller for processing.

   ```
   [edit protocols openflow switch switch-name]
   user@switch# set default-action (drop | packet-in)
   ```

5. (Optional) Configure the **purge-flow-timer** statement, which specifies the number of seconds after which an invalid flow is purged from the flow table.

   ```
   [edit protocols openflow switch switch-name]
   user@switch# set purge-flow-timer seconds
   ```

6. (Optional) Configure OpenFlow traceoptions.

   If you do not configure a log file by specifying its filename, OpenFlow trace messages are logged in the default OpenFlow log file **/var/log/ofd** .

   ```
   [edit protocols openflow]
   user@switch# set traceoptions flag all
   user@switch# set traceoptions file file-name
   ```

RELATED DOCUMENTATION

Example: Enabling OpenFlow on QFX5100 and EX4600 Switches | **102**

Understanding Support for OpenFlow on Devices Running Junos OS | **18**

OpenFlow Operational Mode Commands | **196**

openflow (Protocols OpenFlow) | **183**

Example: Enabling OpenFlow on QFX5100 and EX4600 Switches

IN THIS SECTION

- Requirements | 102
- Overview | 103
- Configuration | 103
- Verification | 106

OpenFlow is an open standard that enables you to control traffic paths in a network by creating, deleting, and modifying flows in each device along a path. This example shows how to configure OpenFlow support on QFX5100 and EX4600 switches.

To isolate and control OpenFlow traffic on the switches, you configure a virtual switch. You also configure a Secure Sockets Layer (SSL) or TCP/IP connection between the virtual switch and a remote OpenFlow controller. Using this connection, the OpenFlow controller can access the flows in the virtual switch.

Requirements

This example uses the following hardware and software components:

- A QFX5100 switch running Junos OS Release 14.1X53-D10 or later, or an EX4600 switch running Junos OS Release 17.1R1 or later.
- An OpenFlow software package is installed on the switch, and the release of this package matches the Junos OS release running on the switch.
- A TCP connection between the switch and an OpenFlow controller.
- A connection between the management interface (em0 or em1) of the switch and the management network.

Overview

In this example, you configure support for OpenFlow on a Juniper Networks switch. The switch has three interfaces that are dedicated to handling OpenFlow traffic: xe-0/0/10.0, xe-0/0/11.0, and xe-0/0/12.0. Note that on these switches, you can configure only a single logical interface, using logical unit number 0 for each OpenFlow interface.

In an OpenFlow topology, a virtual switch is used to isolate and control OpenFlow traffic. You configure the OpenFlow virtual switch and OpenFlow protocol statements at the **[edit protocols openflow]** hierarchy level.

Virtual switch 100 also connects to an OpenFlow controller over a TCP connection at the IP address 10.51.100.174. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow; OpenFlow traffic enters and exits only through these interfaces.

A flow entry consists of a match condition against which packets entering an OpenFlow interface are compared, and the action that is applied to packets that match the condition. Each OpenFlow interface can have one or more flow entries. The **default-action** statement in the OpenFlow configuration indicates the action the switch applies to packets that do not have a matching flow entry. This example uses the **drop** option, which specifies that packets that do not match a flow entry are dropped.

This example also configures OpenFlow traceoptions, along with the **flag all** statement, which captures and logs all OpenFlow events. This example does not configure a specific filename for the log file. As a result, OpenFlow events are logged in the default OpenFlow log directory **/var/log/ofd**.

Configuration

CLI Quick Configuration

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

```
set interfaces xe-0/0/10 unit 0 family ethernet-switching
set interfaces xe-0/0/11 unit 0 family ethernet-switching
set interfaces xe-0/0/12 unit 0 family ethernet-switching
set protocols openflow switch 100 controller address 10.51.100.174
set protocols openflow switch 100 controller protocol tcp
set protocols openflow switch 100 interfaces xe-0/0/10.0
set protocols openflow switch 100 interfaces xe-0/0/11.0
set protocols openflow switch 100 interfaces xe-0/0/12.0
set protocols openflow switch 100 default-action drop
set protocols openflow traceoptions flag all
```

Step-by-Step Procedure

To configure support for OpenFlow:

1. Configure the OpenFlow interfaces as Layer 2 interfaces.

```
[edit interfaces]
user@switch# set xe-0/0/10 unit 0 family ethernet-switching
user@switch# set xe-0/0/11 unit 0 family ethernet-switching
user@switch# set xe-0/0/12 unit 0 family ethernet-switching
```

2. Configure an OpenFlow virtual switch named 100.

```
[edit protocols openflow]
user@switch# set switch 100
```

3. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch 100]
user@switch# set controller address 10.51.100.174
user@switch# set controller protocol tcp
```

4. Configure the logical interfaces in this virtual switch that participate in OpenFlow.

```
[edit protocols openflow switch 100]
user@switch# set interfaces xe-0/0/10.0
user@switch# set interfaces xe-0/0/11.0
user@switch# set interfaces xe-0/0/12.0
```

5. Configure the default action for packets that do not have a matching flow entry.

```
[edit protocols openflow switch 100]
user@switch# set default-action drop
```

6. Configure OpenFlow traceoptions.

```
[edit protocols openflow]
user@switch# set traceoptions flag all
```

7. Commit the configuration.

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

Results

From operational mode, confirm your configuration by entering the **show configuration interfaces**and **show configuration protocols openflow** commands.

```
user@switch> show configuration interfaces
xe-0/0/10 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/0/11 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/0/12 {
    unit 0 {
        family ethernet-switching;
    }
}
```

`user@switch>` **show configuration protocols openflow**

```
switch 100 {
    default-action {
        drop;
    }
    interfaces {
        xe-0/0/10.0;
        xe-0/0/11.0;
        xe-0/0/12.0;
    }
    controller {
        protocol {
            tcp {
        }
```

```
        address 10.51.100.174;
    }
}
traceoptions {
    flag all;
}
```

Verification

IN THIS SECTION

- Verifying That the OpenFlow Controller Connection Is Up | 106
- Verifying that the OpenFlow Interfaces Are Up | 107

Confirm that the configuration is working properly.

Verifying That the OpenFlow Controller Connection Is Up

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command, and verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch automatically initiates a connection to the controller after you commit the configuration.

```
user@switch> show openflow controller
```

```
Openflowd controller information:
Controller socket: 12
Controller IP address: 10.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 4
Controller role: equal
```

```
Negotiated version: 4
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying that the OpenFlow Interfaces Are Up

Purpose

Verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

`user@switch>` **show openflow interfaces**

```
Switch name: 100
Interface Name: xe-0/0/10.0
Interface port number: 41507
Interface Hardware Address: 00:00:5e:00:53:00
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: 100
Interface Name: xe-0/0/11.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:01
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: 100
Interface Name: xe-0/0/12.0
Interface port number: 45549
Interface Hardware Address: 00:00:5e:00:53:02
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
```

```
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Configuring Support for OpenFlow on QFX5100 and EX4600 Switches | **99**
- OpenFlow Operational Mode Commands | **196**
- openflow (Protocols OpenFlow) | **183**

Configuring Support for OpenFlow on EX4550 Switches

IN THIS SECTION

- Configuring the OpenFlow Interfaces | **109**
- Configuring the OpenFlow Protocol | **109**

The following sections configure EX4550 switches to support OpenFlow using interfaces that participate solely in OpenFlow.

Before configuring support for OpenFlow, ensure that the switch meets the following requirements:

- EX4550 switch running Junos OS Release 13.2X51-D20 or a later release
- OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed

- TCP connection between the switch and an OpenFlow controller
- Connection between the management interface of the switch and the management network, which is reachable from the controller IP address

Configuration tasks are described in detail in the following sections:

Configuring the OpenFlow Interfaces

You must configure interfaces participating in OpenFlow as Layer 2 interfaces. On EX Series Ethernet Switches, you configure the interfaces with protocol family **ethernet-switching**. On EX4550 switches, you can configure only a single logical port using logical unit number 0.

To configure the OpenFlow interfaces:

- Configure the logical interface and the protocol family.

```
[edit interfaces interface-name]
user@host# set unit 0 family ethernet-switching
```

Configuring the OpenFlow Protocol

To configure support for OpenFlow, create a virtual switch instance, and specify a switch name, which must be 60 characters or less. For the virtual switch instance, configure the OpenFlow controller information and the participating logical interfaces. Optionally, configure the default action for packets that do not match a flow entry, the purge timer for invalid flows, and any OpenFlow traceoptions.

To configure the OpenFlow protocol:

1. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch switch-name]
user@host# set controller address address
user@host# set controller protocol tcp
```

2. Specify the logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch switch-name]
user@host# set interfaces interface-name1.0
```

```
user@host# set interfaces interface-name2.0
```

3. (Optional) Configure the **default-action** statement for packets that do not match on an existing flow entry.

   If you do not configure the **default-action** statement, the default is **packet-in**, which indicates that packets with no matching flow entry must be sent to the controller for processing.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set default-action (drop | packet-in)
   ```

4. (Optional) Configure the **purge-flow-timer** statement, which is the number of seconds after which an invalid flow is purged from the flow table.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set purge-flow-timer seconds
   ```

5. (Optional) Configure OpenFlow traceoptions.

   If you do not configure a log filename, OpenFlow trace messages are logged in the default OpenFlow log file **/var/log/ofd** .

   ```
   [edit protocols openflow]
   user@host# set traceoptions flag all
   user@host# set traceoptions file file-name
   ```

RELATED DOCUMENTATION

- Example: Enabling OpenFlow on EX4550 Switches | 111
- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- OpenFlow Operational Mode Commands | 196
- openflow (Protocols OpenFlow) | 183

Example: Enabling OpenFlow on EX4550 Switches

IN THIS SECTION

- Requirements | 111
- Overview | 111
- Configuration | 112
- Verification | 115

OpenFlow is an open standard that allows you to control traffic paths in a network by creating, deleting, and modifying flows in each device, including EX4550 switches, along a path. This example shows how to configure OpenFlow support on an EX4550 switch.

Requirements

This example uses the following hardware and software components:

- An EX4550 switch running Junos OS Release 13.2X51-D15 or a later release.
- An OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed.
- A TCP connection between the switch and an OpenFlow controller, which needs to access the data plane of the switch.
- A connection between the me0 interface of the switch and the management network.

Overview

In this example, you configure support for OpenFlow on an EX4550 switch. The switch has three interfaces that are dedicated to handling OpenFlow traffic: xe-0/0/4.0, xe-0/0/5.0, and xe-0/0/6.0. Note that on EX4550 switches, you can only configure a single logical unit using logical unit number 0 for OpenFlow interfaces.

In an OpenFlow topology, a virtual switch is used to isolate and control OpenFlow traffic. You configure the OpenFlow virtual switch and OpenFlow protocol statements at the **[edit protocols openflow]** hierarchy

level. In this example, the virtual switch, 100, is assigned a default VLAN, which acts as a logically separate flood domain. The assignment of the default VLAN to virtual switch 100 is automatic, and no configuration is required to set up the default VLAN.

Virtual switch 100 also connects to the controller over a TCP connection at the IP address 198.51.100.174. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow, and OpenFlow traffic will only enter or exit from these interfaces.

A flow entry consists of a match condition against which packets entering an OpenFlow interface are compared, and the action that is applied to packets that match the condition. Each OpenFlow interface can have one or more flow entries. The **default-action** statement in the OpenFlow configuration indicates the action the switch applies to packets that do not have a matching flow entry. If you omit the **default-action** statement, the default action is **packet-in**, which means that packets that have no matching flow entry are sent to the controller for processing. This example explicitly configures **packet-in** as the default action for packets that do not have a matching flow entry.

This example also configures OpenFlow traceoptions, along with the **flag all** statement, which is meant to capture and log all OpenFlow events. Since the example does not configure a specific filename for the log file, OpenFlow events are logged in the default OpenFlow log directory **/var/log/ofd**.

Configuration

CLI Quick Configuration

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

```
set interfaces xe-0/0/4 unit 0 family ethernet-switching
set interfaces xe-0/0/5 unit 0 family ethernet-switching
set interfaces xe-0/0/6 unit 0 family ethernet-switching
set protocols openflow switch 100 controller address 198.51.100.174
set protocols openflow switch 100 controller protocol tcp
set protocols openflow switch 100 interfaces xe-0/0/4.0
set protocols openflow switch 100 interfaces xe-0/0/5.0
set protocols openflow switch 100 interfaces xe-0/0/6.0
set protocols openflow switch 100 default-action packet-in
set protocols openflow traceoptions flag all
```

Step-by-Step Procedure

To configure support for OpenFlow:

1. Configure the OpenFlow interfaces as Layer 2 interfaces.

```
[edit interfaces]
user@switch# set xe-0/0/4 unit 0 family ethernet-switching
user@switch# set xe-0/0/5 unit 0 family ethernet-switching
user@switch# set xe-0/0/6 unit 0 family ethernet-switching
```

2. Configure an OpenFlow virtual switch.

```
[edit protocols openflow]
user@switch# set switch 100
```

3. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch 100]
user@switch# set controller address 198.51.100.174
user@switch# set controller protocol tcp
```

4. Configure the logical interfaces participating in OpenFlow under this virtual switch.

```
[edit protocols openflow switch 100]
user@switch# set interfaces xe-0/0/4.0
user@switch# set interfaces xe-0/0/5.0
user@switch# set interfaces xe-0/0/6.0
```

5. Configure the default action for packets that do not have a matching flow entry.

```
[edit protocols openflow switch 100]
user@switch# set default-action packet-in
```

6. Configure OpenFlow traceoptions.

```
[edit protocols openflow]
user@switch# set traceoptions flag all
```

7. Commit the configuration.

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

Results

From operational mode, confirm your configuration by entering the **show configuration interfaces** and **show configuration protocols openflow** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch> show configuration interfaces
xe-0/0/4 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/0/5 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/0/6 {
    unit 0 {
        family ethernet-switching;
    }
}
```

```
user@switch> show configuration protocols openflow
switch 100 {
    default-action packet-in;
    interfaces {
        xe-0/0/4.0;
        xe-0/0/5.0;
        xe-0/0/6.0;
    }
    controller {
        address 198.51.100.174;
        protocol tcp;
    }
}
traceoptions {
```

```
    flag all;
}
```

Verification

IN THIS SECTION

- Verifying that the OpenFlow Controller Connection is Up | 115
- Verifying that the OpenFlow Interfaces Are Up | 116

Confirm that the configuration is working properly.

Verifying that the OpenFlow Controller Connection is Up

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command, and verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch should automatically initiate a connection to the controller after you commit the configuration.

```
user@switch> show openflow controller
```

```
Openflowd controller information:
Controller socket: 12
Controller IP address: 198.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 4
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying that the OpenFlow Interfaces Are Up

Purpose

Verify that the OpenFlow interfaces are in the **Up** state.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

`user@switch>` **show openflow interfaces**

```
Switch name: 100
Interface Name: xe-0/0/4.0
Interface port number: 41507
Interface Hardware Address: 00:00:5e:00:53:00
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: 100
Interface Name: xe-0/0/5.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:01
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: 100
Interface Name: xe-0/0/6.0
Interface port number: 45549
Interface Hardware Address: 00:00:5e:00:53:02
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of the OpenFlow interfaces is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Configuring Support for OpenFlow on EX4550 Switches | **108**
- OpenFlow Operational Mode Commands | **196**
- openflow (Protocols OpenFlow) | **183**

3

CHAPTER

Configuring OpenFlow Hybrid Interfaces

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | **119**

Configuring OpenFlow Hybrid Interfaces on MX Series Routers | **120**

Example: Configuring OpenFlow Hybrid Interfaces on MX Series Routers | **124**

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | **134**

Example: Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | **138**

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS

On Juniper Networks EX9200 Ethernet Switches and on MX Series 5G Universal Routing Platforms that support OpenFlow, you can configure physical interfaces that support multiple logical interfaces as hybrid interfaces. A hybrid interface concurrently supports OpenFlow logical interfaces and non-OpenFlow logical interfaces.

On a hybrid interface, the OpenFlow protocol and the non-OpenFlow protocols essentially exist independently. Traffic does not get forwarded across OpenFlow and non-OpenFlow logical interfaces. Instead VLANs and VLAN tags are used to distinguish the OpenFlow traffic from the normal traffic. To accomplish this, you must enable the reception and transmission of 802.1Q VLAN-tagged frames on all interfaces, including both hybrid and non-hybrid interfaces. You must also configure separate virtual switch routing instances for OpenFlow traffic and for normal traffic, which serve to separate the VLAN ID space.

On devices using hybrid interfaces, traffic entering an interface must be VLAN-tagged. The VLAN ID differentiates the OpenFlow traffic from the normal traffic, and on the hybrid interface, the VLAN ID also determines the associated logical interface. Once the logical interface is known, the traffic is forwarded accordingly. The device forwards OpenFlow traffic according to OpenFlow flow entries, and it forwards normal traffic using traditional Layer 2 and Layer 3 processing. If you do not configure a native VLAN, untagged packets are dropped.

On a hybrid interface, you configure a logical interface as a trunk interface, which accepts and forwards tagged packets from multiple VLANs. Additionally, you can configure certain non-OpenFlow logical interfaces as Layer 3 subinterfaces that perform traditional Layer 3 or MPLS-based forwarding.

To configure a logical interface to receive and forward VLAN-tagged frames, you must bind a VLAN ID, or a range or list of VLAN IDs, to the logical interface. OpenFlow interfaces must have a different set of VLANs from normal interfaces. On a hybrid interface, OpenFlow traffic can only exit from an interface that has the same VLAN ID range as that of the ingress interface.

A hybrid interface configuration with multiple logical interfaces permits OpenFlow and non-OpenFlow traffic to traverse the same interface while keeping the traffic in separate routing or bridging domains. One advantage of using hybrid interfaces is that you can use fewer physical interfaces where port density is an issue. However, using hybrid interfaces requires some additional configuration, and untagged traffic entering a hybrid port cannot be forwarded according to OpenFlow flow entries. Additionally, several physical port properties such as Layer 1 statistics are reported for all logical interfaces on that physical interface. Thus, when you configure a physical interface in hybrid mode, these properties are reported for all OpenFlow and non-OpenFlow logical interfaces on that physical interface. These properties include queue drops, framing errors, CRC errors, and collisions. When using hybrid interfaces, if you use the Link Layer Discovery Protocol (LLDP) for topology discovery, you must ensure that any LLDP frames entering a hybrid interface are tagged appropriately.

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Configuring OpenFlow Hybrid Interfaces on MX Series Routers | 120

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | 134

Binding VLAN IDs to Logical Interfaces

Configuring OpenFlow Hybrid Interfaces on MX Series Routers

IN THIS SECTION

- Configuring the Hybrid Physical Interface | 121
- Configuring the Hybrid Interface Logical Units | 121
- Configuring the Non-Hybrid Interfaces | 122
- Configuring OpenFlow | 122
- Configuring the Virtual Switch Routing Instances | 123

On MX Series routers that support OpenFlow, you can configure a physical interface as a hybrid interface that concurrently supports OpenFlow logical interfaces and non-OpenFlow logical interfaces. If you configure an OpenFlow hybrid interface on a device running Junos OS, you must enable the reception and transmission of 802.1Q VLAN-tagged frames on all interfaces, including both hybrid and non-hybrid interfaces, and you must configure a virtual switch routing instance for the OpenFlow traffic and a separate virtual switch routing instance for the normal traffic.

The following sections detail configuring an MX Series router that supports OpenFlow with a mix of hybrid and normal interfaces:

Configuring the Hybrid Physical Interface

To configure the hybrid physical interface:

1. Enable VLAN tagging.

   Configure **vlan-tagging** to support 802.1Q VLAN single-tag frames for both OpenFlow and non-OpenFlow traffic, or configure **flexible-vlan-tagging** to support both 802.1Q VLAN single-tag and dual-tag frames.

   ```
   [edit interfaces interface-name]
   user@host# set (vlan-tagging | flexible-vlan-tagging)
   ```

2. Configure flexible Ethernet services encapsulation to enable multiple per-unit Ethernet encapsulations.

   ```
   [edit interfaces interface-name]
   user@host# set encapsulation flexible-ethernet-services
   ```

Configuring the Hybrid Interface Logical Units

On a hybrid interface, you configure an OpenFlow or non-OpenFlow logical interface as a Layer 2 trunk interface. Additionally, you can configure a non-OpenFlow logical interface as a Layer 3 subinterface that performs traditional Layer 3 or MPLS-based forwarding. To configure a logical interface to receive and forward VLAN-tagged frames, you must bind a VLAN ID, or a range or list of VLAN IDs, to the logical interface. Configure Layer 2 interfaces using family **bridge** on MX Series routers.

To configure the hybrid interface logical units:

1. Configure the OpenFlow logical interfaces and any non-OpenFlow Layer 2 logical interfaces, and specify the interface mode and VLAN membership.

   ```
   [edit interfaces interface-name]
   user@host# set unit unit family bridge interface-mode trunk
   user@host# set unit unit family bridge vlan-id-list vlan-ids
   ```

2. Configure any non-OpenFlow Layer 3 logical interfaces, and specify the VLAN membership.

   ```
   [edit interfaces interface-name]
   ```

```
user@host# set unit unit (vlan-id | vlan-id-list | vlan-id-range) vlan-ids
user@host# set unit unit family inet address address
```

Configuring the Non-Hybrid Interfaces

Non-hybrid interfaces support either OpenFlow traffic or non-OpenFlow traffic, but not both simultaneously.

To configure the non-hybrid interfaces:

1. Configure interfaces that support only OpenFlow traffic as Layer 2 trunk interfaces, and specify the interface mode and VLAN membership.

   ```
   [edit interfaces interface-name]
   user@host# set vlan-tagging
   user@host# set unit unit family bridge interface-mode trunk
   user@host# set unit unit family bridge vlan-id-list vlan-ids
   ```

2. Configure interfaces that support only normal traffic, and specify the interface mode for the Layer 2 interfaces and the VLAN membership.

   For example:

   ```
   [edit interfaces interface-name]
   user@host# set vlan-tagging
   user@host# set unit unit family bridge interface-mode trunk
   user@host# set unit unit family bridge vlan-id-list vlan-ids
   ```

Configuring OpenFlow

To configure the OpenFlow virtual switch instance:

1. Configure the OpenFlow controller IP address and the connection protocol.

   ```
   [edit protocols openflow switch switch-name]
   user@host# set controller address address
   user@host# set controller protocol tcp port port
   ```

2. Specify all logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch switch-name]
user@host# set interfaces interface-name
```

Configuring the Virtual Switch Routing Instances

Configure separate virtual switch routing instances for the OpenFlow traffic and the non-OpenFlow traffic. The configured interface names must include a logical unit number.

To configure the virtual switch routing instances:

1. Configure the virtual switch routing instance for the OpenFlow traffic, and specify the OpenFlow logical interfaces and VLANs.

```
[edit routing-instances of-routing-instance-name]
user@host# set instance-type virtual-switch
user@host# set interface of-interface-name1
user@host# set interface of-interface-name2
user@host# set bridge-domains name vlan-id-list of-vlan-id-list
```

2. Configure the virtual switch routing instance for the non-OpenFlow traffic, and specify the non-OpenFlow logical interfaces and VLANs.

```
[edit routing instances non-of-routing-instance-name]
user@host# set instance-type virtual-switch
user@host# set interface non-of-interface-name1
user@host# set interface non-of-interface-name2
user@host# set bridge-domains name vlan-id-list non-of-vlan-id-list
```

RELATED DOCUMENTATION

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | 119

Example: Configuring OpenFlow Hybrid Interfaces on MX Series Routers | 124

Example: Configuring OpenFlow Hybrid Interfaces on MX Series Routers

IN THIS SECTION

- Requirements | 124
- Overview | 125
- Configuration | 126
- Verification | 132

On MX series routers that support OpenFlow, you can configure physical interfaces that support multiple logical interfaces as hybrid interfaces. A hybrid interface concurrently supports both OpenFlow logical interfaces and non-OpenFlow logical interfaces, thus allowing OpenFlow and non-OpenFlow traffic to traverse the same interface.

Hybrid interfaces enable you to use your physical interfaces more efficiently, especially in a situation where port density is an issue.

This example shows how to configure an MX Series router with OpenFlow hybrid interfaces.

Requirements

This example uses the following hardware and software components:

- MX240 router running Junos OS Release 13.3 or a later release
- OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed
- TCP connection between the router and an OpenFlow controller
- Connection between the fxp0 management interface of the router and the management network, which is reachable from the OpenFlow controller IP address

Overview

In this example, you configure an MX240 router with a hybrid interface, ge-1/0/1, an OpenFlow interface, ge-1/0/2, and a non-OpenFlow interface, ge-1/0/3. On the hybrid interface, logical interface ge-1/0/1.0 participates in OpenFlow, and logical interfaces ge-1/0/1.1 and ge-1/0/1.2 do not participate in OpenFlow.

When using OpenFlow hybrid interfaces, you use VLANs to distinguish the OpenFlow traffic from the normal traffic. Thus, you must enable VLAN tagging on all interfaces, and traffic entering the interfaces must be vlan-tagged. Untagged traffic entering the hybrid interface is dropped. In this example, you configure the hybrid interface using **flexible-vlan-tagging**, which enables VLAN tagging and supports both 802.1Q VLAN single-tag and dual-tag frames for all traffic on the interface. You configure interfaces ge-1/0/2 and ge-1/0/3 using **vlan-tagging**.

You configure the hybrid interface encapsulation as flexible Ethernet services. Note that for interfaces with this encapsulation, all VLAN IDs are valid. VLAN IDs from 1 through 511 are no longer reserved for normal VLANs. In this example, VLANs 1 through 100 are used for OpenFlow traffic, and VLANs 101 through 200 and VLAN 300 are used for normal traffic.

All logical interfaces except ge-1/0/1.2 are configured as Layer 2 trunk interfaces using family **bridge** and interface mode **trunk**. Logical interfaces ge-1/0/1.0 and ge-1/0/2.0 participate in OpenFlow and receive and forward traffic with OpenFlow VLAN IDs 1 through 100. Logical interfaces ge-1/0/1.1 and ge-1/0/3.0 do not participate in OpenFlow and receive and forward traffic with non-OpenFlow VLAN IDs 101 through 200.

ge-1/0/1.2 is a Layer 3 logical interface with IP address 198.51.100.10/24 that performs Layer 3 routing. This interface does not participate in OpenFlow and routes traffic with VLAN ID 300.

Table 42 on page 125 summarizes the logical interfaces, traffic type, and associated VLAN IDs.

Table 42: Summary of Logical Interfaces

Logical Interface	Traffic Type	VLANs
ge-1/0/1.0	OpenFlow	1 through 100
ge-1/0/1.1	non-OpenFlow	101 through 200
ge-1/0/1.2	non-OpenFlow	300
ge-1/0/2.0	OpenFlow	1 through 100
ge-1/0/3.0	non-OpenFlow	101 through 200

You configure the OpenFlow virtual switch and OpenFlow protocol statements at the **[edit protocols openflow]** hierarchy level. The virtual switch, OFswitch2, connects to the controller over a TCP connection

at IP address 172.16.1.1. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow, which includes ge-1/0/1.0 and ge-1/0/2.0.

When configuring OpenFlow on MX Series routers, you must configure a virtual switch routing instance for the OpenFlow traffic that isolates it from the normal network traffic. Additionally, when using hybrid interfaces, you configure both a virtual switch routing instance for the OpenFlow traffic and also a separate virtual switch routing instance for the normal traffic. In this example, you configure routing instance rt1 for the OpenFlow traffic and routing instance rt2 for the normal traffic.

Routing instance rt1 includes the interfaces participating in OpenFlow, ge-1/0/1.0 and ge-1/0/2.0. Within the routing instance you configure the bridge domain to include all OpenFlow VLANs 1 through 100. Routing instance rt2 includes the Layer 2 interfaces that do not participate in OpenFlow, ge-1/0/1.1 and ge-1/0/3.0. Within the routing instance you configure the bridge domain to include the non-OpenFlow VLANs 101 through 200.

> **NOTE:** In order to direct OpenFlow traffic, the OpenFlow controller must install flow entries that select the appropriate traffic and forward it to the correct OpenFlow interface.

Configuration

IN THIS SECTION

- Configuring the Interfaces | **127**
- Configuring OpenFlow | **128**
- Configuring the Virtual Switch Routing Instances | **129**
- Results | **129**

CLI Quick Configuration

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

```
set interfaces ge-1/0/1 flexible-vlan-tagging
set interfaces ge-1/0/1 encapsulation flexible-ethernet-services
set interfaces ge-1/0/1 unit 0 family bridge interface-mode trunk
```

```
set interfaces ge-1/0/1 unit 0 family bridge vlan-id-list 1-100
set interfaces ge-1/0/1 unit 1 family bridge interface-mode trunk
set interfaces ge-1/0/1 unit 1 family bridge vlan-id-list 101-200
set interfaces ge-1/0/1 unit 2 vlan-id 300
set interfaces ge-1/0/1 unit 2 family inet address 198.51.100.10/24
set interfaces ge-1/0/2 vlan-tagging
set interfaces ge-1/0/2 unit 0 family bridge interface-mode trunk
set interfaces ge-1/0/2 unit 0 family bridge vlan-id-list 1-100
set interfaces ge-1/0/3 vlan-tagging
set interfaces ge-1/0/3 unit 0 family bridge interface-mode trunk
set interfaces ge-1/0/3 unit 0 family bridge vlan-id-list 101-200
set protocols openflow switch OFswitch2 controller address 172.16.1.1
set protocols openflow switch OFswitch2 controller protocol tcp port 6633
set protocols openflow switch OFswitch2 interfaces ge-1/0/1.0
set protocols openflow switch OFswitch2 interfaces ge-1/0/2.0
set routing-instances rt1 instance-type virtual-switch
set routing-instances rt1 interface ge-1/0/1.0
set routing-instances rt1 interface ge-1/0/2.0
set routing-instances rt1 bridge-domains bd-of vlan-id-list 1-100
set routing-instances rt2 instance-type virtual-switch
set routing-instances rt2 interface ge-1/0/1.1
set routing-instances rt2 interface ge-1/0/3.0
set routing-instances rt2 bridge-domains bd-nonof vlan-id-list 101-200
```

Configuring the Interfaces

Step-by-Step Procedure

To configure the interfaces:

1. On the hybrid physical interface, enable VLAN tagging and configure the encapsulation.

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

2. Configure OpenFlow logical interface ge-1/0/1.0 as a Layer 2 trunk that supports VLANs 1-100.

```
[edit interfaces ge-1/0/1]
user@host# set unit 0 family bridge interface-mode trunk
user@host# set unit 0 family bridge vlan-id-list 1-100
```

3. Configure non-OpenFlow logical interface ge-1/0/1.1 as a Layer 2 trunk that supports VLANs 101-200.

```
[edit interfaces ge-1/0/1]
user@host# set unit 1 family bridge interface-mode trunk
user@host# set unit 1 family bridge vlan-id-list 101-200
```

4. Configure non-OpenFlow logical interface ge-1/0/1.2 as a Layer 3 subinterface.

```
[edit interfaces ge-1/0/1]
user@host# set unit 2 vlan-id 300
user@host# set unit 2 family inet address 198.51.100.10/24
```

5. On ge-1/0/2, enable VLAN tagging, and configure the logical interface as a Layer 2 trunk that supports VLANs 1-100.

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

6. On ge-1/0/3, enable VLAN tagging, and configure the logical interface as a Layer 2 trunk that supports VLANs 101-200:

```
[edit interfaces ge-1/0/3]
user@host# set vlan-tagging
user@host# set unit 0 family bridge interface-mode trunk
user@host# set unit 0 family bridge vlan-id-list 101-200
```

Configuring OpenFlow

Step-by-Step Procedure

To configure OpenFlow:

1. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch OFswitch2]
user@host# set controller address 172.16.1.1
user@host# set controller protocol tcp port 6633
```

2. Specify the logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch OFswitch2]
user@host# set interfaces ge-1/0/1.0
user@host# set interfaces ge-1/0/2.0
```

Configuring the Virtual Switch Routing Instances

Step-by-Step Procedure

To configure the virtual switch routing instances:

1. Configure the virtual switch routing instance for the OpenFlow traffic.

```
[edit]
user@host# set routing-instances rt1 instance-type virtual-switch
user@host# set routing-instances rt1 interface ge-1/0/1.0
user@host# set routing-instances rt1 interface ge-1/0/2.0
user@host# set routing-instances rt1 bridge-domains bd-of vlan-id-list 1-100
```

2. Configure the virtual switch routing instance for the non-OpenFlow traffic.

```
[edit]
user@host# set routing-instances rt2 instance-type virtual-switch
user@host# set routing-instances rt2 interface ge-1/0/1.1
user@host# set routing-instances rt2 interface ge-1/0/3.0
user@host# set routing-instances rt2 bridge-domains bd-nonof vlan-id-list 101-200
```

3. Commit the configuration.

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

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols openflow**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
user@host# show interfaces
```

```
ge-1/0/1 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 1-100;
        }
    }
    unit 1 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 101-200;
        }
    }
    unit 2 {
        vlan-id 300;
        family inet {
            address 198.51.100.10/24;
        }
    }
}

ge-1/0/2 {
vlan-tagging;
unit 0 {
    family bridge {
       interface-mode trunk;
        vlan-id-list 1-100;
    }
}

ge-1/0/3 {
vlan-tagging;
unit 0 {
    family bridge {
        interface-mode trunk;
        vlan-id-list 101-200;
    }
}
```

```
user@host# show protocols openflow
```

```
switch OFswitch2 {
    interfaces {
        ge-1/0/1.0;
        ge-1/0/2.0;
    }
    controller {
        protocol tcp {
            port 6633;
        }
        address 172.16.1.1;
    }
}
```

user@host# **show routing-instances**

```
rt1 {
    instance-type virtual-switch;
    interface ge-1/0/1.0;
    interface ge-1/0/2.0;
    bridge-domains {
        bd-of {
            vlan-id-list 1-100;
        }
    }
}
rt2 {
    instance-type virtual-switch;
    interface ge-1/0/1.1;
    interface ge-1/0/3.0;
    bridge-domains {
        bd-nonof {
            vlan-id-list 101-200;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying that the OpenFlow Controller Connection is Up | 132
- Verifying that the OpenFlow Interfaces Are Up | 132

Confirm that the configuration is working properly.

Verifying that the OpenFlow Controller Connection is Up

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command, and verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch should automatically initiate a connection to the controller after you commit the configuration.

```
user@host> show openflow controller
```

```
Openflowd controller information:
Controller socket: 11
Controller IP address: 172.16.1.1
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 1
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying that the OpenFlow Interfaces Are Up

Purpose

Verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

```
user@host> show openflow interfaces
```

```
Switch name: OFswitch2
Interface Name: ge-1/0/1.0
Interface port number: 41500
Interface Hardware Address: 00:00:5e:00:53:a1
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch2
Interface Name: ge-1/0/2.0
Interface port number: 41501
Interface Hardware Address: 00:00:5e:00:53:00
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | 119

Configuring OpenFlow Hybrid Interfaces on MX Series Routers | 120

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches

IN THIS SECTION

- Configuring the Hybrid Physical Interface | 134
- Configuring the Hybrid Interface Logical Units | 135
- Configuring the Non-Hybrid Interfaces | 135
- Configuring OpenFlow | 136
- Configuring the Virtual Switch Routing Instances | 137

On EX9200 switches that support OpenFlow, you can configure a physical interface as a hybrid interface that concurrently supports OpenFlow logical interfaces and non-OpenFlow logical interfaces. If you configure an OpenFlow hybrid interface on a device running Junos OS, you must enable the reception and transmission of 802.1Q VLAN-tagged frames on all interfaces, including both hybrid and non-hybrid interfaces, and you must configure a virtual switch routing instance for the OpenFlow traffic and a separate virtual switch routing instance for the normal traffic.

The following sections detail configuring an EX9200 switch that supports OpenFlow with a mix of hybrid and normal interfaces:

Configuring the Hybrid Physical Interface

To configure the hybrid physical interface:

1. Enable VLAN tagging.

   Configure **vlan-tagging** to support 802.1Q VLAN single-tag frames for both OpenFlow and non-OpenFlow traffic, or configure **flexible-vlan-tagging** to support both 802.1Q VLAN single-tag and dual-tag frames.

   ```
   [edit interfaces interface-name]
   user@host# set (vlan-tagging | flexible-vlan-tagging)
   ```

2. Configure flexible Ethernet services encapsulation to enable multiple per-unit Ethernet encapsulations.

```
[edit interfaces interface-name]
user@host# set encapsulation flexible-ethernet-services
```

Configuring the Hybrid Interface Logical Units

On a hybrid interface, you configure an OpenFlow or non-OpenFlow logical interface as a Layer 2 trunk interface. Additionally, you can configure a non-OpenFlow logical interface as a Layer 3 subinterface that performs traditional Layer 3 forwarding. To configure a logical interface to receive and forward VLAN-tagged frames, you must bind a VLAN ID, or a range or list of VLAN IDs, to the logical interface. Configure Layer 2 interfaces using family **ethernet-switching** on EX9200 switches.

To configure the hybrid interface logical units:

1. Configure the OpenFlow logical interfaces and any non-OpenFlow Layer 2 logical interfaces, and specify the interface mode and VLAN membership.

```
[edit interfaces interface-name]
user@host# set unit unit family ethernet-switching interface-mode trunk
user@host# set unit unit family ethernet-switching vlan members vlan-ids
```

2. Configure any non-OpenFlow Layer 3 logical interfaces, and specify the VLAN membership.

```
[edit interfaces interface-name]
user@host# set unit unit (vlan-id | vlan-id-list | vlan-id-range) vlan-ids
user@host# set unit unit family inet address address
```

Configuring the Non-Hybrid Interfaces

Non-hybrid interfaces support either OpenFlow traffic or non-OpenFlow traffic, but not both simultaneously.

To configure the non-hybrid interfaces:

1. Configure interfaces that support only OpenFlow traffic as Layer 2 trunk interfaces, and specify the interface mode and VLAN membership.

```
[edit interfaces interface-name]
user@host# set vlan-tagging
user@host# set unit unit family ethernet-switching interface-mode trunk
user@host# set unit unit family ethernet-switching vlan members (vlan-id | vlan-id-list)
```

2. Configure interfaces that support only normal traffic, and specify the interface mode for the Layer 2 interfaces and the VLAN membership.

   For example:

```
[edit interfaces interface-name]
user@host# set vlan-tagging
user@host# set unit unit family ethernet-switching interface-mode trunk
user@host# set unit unit family ethernet-switching vlan members (vlan-id | vlan-id-list)
```

Configuring OpenFlow

To configure the OpenFlow virtual switch instance:

1. Configure the OpenFlow controller IP address and the connection protocol.

```
[edit protocols openflow switch switch-name]
user@host# set controller address address
user@host# set controller protocol tcp port port
```

2. Specify all logical interfaces participating in OpenFlow under this virtual switch instance.

```
[edit protocols openflow switch switch-name]
user@host# set interfaces interface-name
```

Configuring the Virtual Switch Routing Instances

Configure separate virtual switch routing instances for the OpenFlow traffic and the non-OpenFlow traffic. The configured interface names must include a logical unit number.

To configure the virtual switch routing instances:

1. Configure the virtual switch routing instance for the OpenFlow traffic, and specify the OpenFlow logical interfaces and VLANs.

```
[edit routing-instances of-routing-instance-name]
user@host# set instance-type virtual-switch
user@host# set interface of-interface-name1
user@host# set interface of-interface-name2
user@host# set vlans name vlan-id-list of-vlan-id-list
```

2. Configure the virtual switch routing instance for the non-OpenFlow traffic, and specify the non-OpenFlow logical interfaces and VLANs.

```
[edit routing instances non-of-routing-instance-name]
user@host# set instance-type virtual-switch
user@host# set interface non-of-interface-name1
user@host# set interface non-of-interface-name2
user@host# set vlans name vlan-id-list non-of-vlan-id-list
```

RELATED DOCUMENTATION

Example: Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | 138

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | 119

Example: Configuring OpenFlow Hybrid Interfaces on EX9200 Switches

IN THIS SECTION

- Requirements | 138
- Overview and Topology | 139
- Configuration | 140
- Verification | 146

On EX9200 switches that have the OpenFlow software package installed, you can configure physical interfaces that support multiple logical interfaces as OpenFlow hybrid interfaces. A hybrid interface concurrently supports OpenFlow logical interfaces and non-OpenFlow logical interfaces. A hybrid interface enables OpenFlow and non-OpenFlow traffic to traverse the same physical interface while keeping the traffic in separate VLANs.

Hybrid interfaces enable you to use physical interfaces more efficiently, especially in a situation where having an adequate number of physical interfaces available is important.

This example shows how to configure an OpenFlow hybrid interface on an EX9200 switch.

Requirements

This example uses the following hardware and software components:

- An EX9200 switch running Junos OS Release 13.3 or a later release.
- An OpenFlow software package is installed on the switch, and the software package release matches the Junos OS release running on the switch.
- The switch has a TCP connection to an OpenFlow controller, which needs to access the data plane of the switch.
- The switch is connected to the management network through the fxp0 interface and is reachable from the controller IP address.

Overview and Topology

In this example, you configure an EX9200 switch with:

- One hybrid interface, xe-2/1/0
- One non-hybrid interface, xe-2/1/1, which handles OpenFlow traffic only
- One non-hybrid interface, xe-2/1/2, which handles non-OpenFlow traffic only

On the hybrid interface, logical interface xe-2/1/0.0 participates in OpenFlow, and logical interfaces xe-2/1/0.1 and xe-2/1/0.2 do not participate in OpenFlow.

When using hybrid interfaces, you use VLAN tagging to distinguish OpenFlow traffic from non-OpenFlow traffic. Thus, you must enable VLAN tagging on all interfaces, and traffic entering the interfaces must be VLAN-tagged. If you do not configure a native VLAN, untagged traffic entering a hybrid interface is dropped. In this example, you configure the hybrid interface by using **flexible-vlan-tagging**, which enables VLAN tagging and supports both 802.1Q VLAN single-tag and dual-tag frames for all traffic on the interface. You also configure the OpenFlow interface xe-2/1/1 and the non-OpenFlow interface xe-2/1/2 by using **vlan-tagging**, which enables VLAN tagging and supports only 802.1Q VLAN single-tag frames for all traffic on the interface.

You configure the hybrid interface encapsulation as flexible Ethernet services. Note that for interfaces with this type of encapsulation, all VLAN IDs are valid. VLAN IDs from 1 through 511 are no longer reserved for normal Ethernet VLANs. In this example, VLANs 100 through 200 are used for OpenFlow traffic, and VLANs 700 and 800 are used for non-OpenFlow traffic.

All logical interfaces except xe-2/1/0.2 are configured as Layer 2 trunk interfaces by using family **ethernet-switching** and interface mode **trunk**. Logical interfaces xe-2/1/0.0 and xe-2/1/1.0 participate in OpenFlow and receive and forward traffic with OpenFlow VLAN IDs 100 through 200. Logical interfaces xe-2/1/0.1 and xe-2/1/2.0 do not participate in OpenFlow and receive and forward traffic with non-OpenFlow VLAN ID 700.

Logical interface xe-2/1/0.2 is a subinterface with the IP address 198.51.100.10/24 and performs Layer 3 routing. This interface does not participate in OpenFlow and routes traffic with VLAN ID 800.

Table 43 on page 139 summarizes the logical interfaces, traffic types, and associated VLAN IDs.

Table 43: Summary of Logical Interface Configuration in EX9200 Hybrid Interface Example

Logical Interface	Traffic Type	VLANs
xe-2/1/0.0	OpenFlow	100 through 200
xe-2/1/0.1	Non-OpenFlow	700
xe-2/1/0.2	Non-OpenFlow	800

Table 43: Summary of Logical Interface Configuration in EX9200 Hybrid Interface Example *(continued)*

Logical Interface	Traffic Type	VLANs
xe-2/1/1.0	OpenFlow	200
xe-2/1/2.0	Non-OpenFlow	700

You configure the OpenFlow virtual switch and OpenFlow protocol statements at the **[edit protocols openflow]** hierarchy level. The virtual switch 100 connects to the OpenFlow controller over a TCP connection at the IP address 198.51.100.174. The virtual switch configuration must include all of the logical interfaces participating in OpenFlow, which includes xe-2/1/0.0 and xe-2/1/1.0.

An EX9200 switch requires a separate routing instance for a virtual switch. This routing instance isolates the OpenFlow traffic from the non-OpenFlow traffic. When using hybrid interfaces, you configure a virtual switch routing instance for the OpenFlow traffic and another virtual switch routing instance for non-OpenFlow traffic. In this example, you configure routing instance **OF** for the OpenFlow traffic and routing instance **NON-OF** for the non-OpenFlow traffic.

Routing instance **OF** includes the interfaces participating in OpenFlow—xe-2/1/0.0 and xe-2/1/1.0. Within this routing instance, you configure a VLAN to include OpenFlow VLANs 100 through 200. Routing instance **NON-OF** includes the Layer 2 interfaces that do not participate in OpenFlow—xe-2/1/0.1 and xe-2/1/2.0. Within this routing instance, you configure a VLAN to include the non-OpenFlow VLAN 700.

> **NOTE:** To direct OpenFlow traffic, the OpenFlow controller must install flow entries that select the appropriate traffic and forward it to the correct OpenFlow interface.

Configuration

IN THIS SECTION

- Configuring the Interfaces | **141**
- Configuring OpenFlow | **142**
- Configuring the Virtual Switch Routing Instances | **143**
- Results | **144**

CLI Quick Configuration

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

```
set interfaces xe-2/1/0 flexible-vlan-tagging
set interfaces xe-2/1/0 encapsulation flexible-ethernet-services
set interfaces xe-2/1/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-2/1/0 unit 0 family ethernet-switching vlan members 100-200
set interfacesxe-2/1/0 unit 1 family ethernet-switching interface-mode trunk
set interfaces xe-2/1/0 unit 1 family ethernet-switching vlan members 700
set interfaces xe-2/1/0 unit 2 vlan-id 800
set interfaces xe-2/1/0 unit 2 family inet address 198.51.100.10/24
set interfaces xe-2/1/1 vlan-tagging
set interfaces xe-2/1/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-2/1/1 unit 0 family ethernet-switching vlan members 200
set interfaces xe-2/1/2 vlan-tagging
set interfaces xe-2/1/2 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-2/1/2 unit 0 family ethernet-switching vlan members 700
set protocols openflow switch 100 controller address 198.51.100.174
set protocols openflow switch 100 controller protocol tcp port 6633
set protocols openflow switch 100 interfaces xe-2/1/0.0
set protocols openflow switch 100 interfaces xe-2/1/1.0
set routing-instances OF instance-type virtual-switch
set routing-instances OF interface xe-2/1/0.0
set routing-instances OF interface xe-2/1/1.0
set routing-instances OF vlans OF-vlan vlan-id-list 100-200
set routing-instances NON-OF instance-type virtual-switch
set routing-instances NON-OF interface xe-2/1/0.1
set routing-instances NON-OF interface xe-2/1/2.0
set routing-instances NON-OF vlans OF-vlan vlan-id-list 700
```

Configuring the Interfaces

Step-by-Step Procedure

To configure the interfaces:

1. On the hybrid physical interface, enable VLAN tagging and configure the encapsulation:

```
[edit interfaces xe-2/1/0]
user@switch# set flexible-vlan-tagging
user@switch# set encapsulation flexible-ethernet-services
```

2. Configure the OpenFlow logical interface xe-2/1/0.0 as a Layer 2 trunk that supports VLANs 100 through 200:

```
[edit interfaces xe-2/1/0]
user@switch# set unit 0 family ethernet-switching interface-mode trunk
user@switch# set unit 0 family ethernet-switching vlan members 100-200
```

3. Configure the non-OpenFlow logical interface xe-2/1/0.1 as a Layer 2 trunk that supports VLAN 700:

```
[edit interfaces xe-2/1/0]
user@switch# set unit 1 family ethernet-switching interface-mode trunk
user@switch# set unit 1 family ethernet-switching vlan members 700
```

4. Configure the non-OpenFlow logical interface xe-2/1/0.2 as a Layer 3 subinterface:

```
[edit interfaces xe-2/1/0]
user@switch# set unit 2 vlan-id 800
user@switch# set unit 2 family inet address 198.51.100.10/24
```

5. On xe-2/1/1, enable VLAN tagging, and configure the logical interface as a Layer 2 trunk that supports VLAN 200:

```
[edit interfaces xe-2/1/1]
user@switch# set vlan-tagging
user@switch# set unit 0 family ethernet-switching interface-mode trunk
user@switch# set unit 0 family ethernet-switching vlan members 200
```

6. On xe-2/1/2, enable VLAN tagging, and configure the logical interface as a Layer 2 trunk that supports VLAN 700:

```
[edit interfaces xe-2/1/2]
user@switch# set vlan-tagging
user@switch# set unit 0 family ethernet-switching interface-mode trunk
user@switch# set unit 0 family ethernet-switching vlan members 700
```

Configuring OpenFlow

Step-by-Step Procedure

To configure OpenFlow:

1. Configure the OpenFlow controller IP address and the connection protocol:

```
[edit protocols openflow switch 100]
user@switch# set controller address 198.51.100.174
user@switch# set controller protocol tcp port 6633
```

2. Specify the logical interfaces participating in OpenFlow under virtual switch 100:

```
[edit protocols openflow switch 100]
user@switch# set interfaces xe-2/1/0.0
user@switch# set interfaces xe-2/1/1.0
```

Configuring the Virtual Switch Routing Instances

Step-by-Step Procedure

To configure the routing instances:

1. Configure the routing instance for the OpenFlow traffic.

```
[edit]
user@switch# set routing-instances OF instance-type virtual-switch
user@switch# set routing-instances OF interface xe-2/1/0.0
user@switch# set routing-instances OF interface xe-2/1/1.0
user@switch# set routing-instances OF vlans OF-vlan vlan-id-list 100-200
```

2. Configure the routing instance for the non-OpenFlow traffic on Layer 2 interfaces:

```
[edit]
user@switch# set routing-instances NON-OF instance-type virtual-switch
user@switch# set routing-instances NON-OF interface xe-2/1/0.1
user@switch# set routing-instances NON-OF interface xe-2/1/2.0
user@switch# set routing-instances NON-OF vlans NOF-vlan vlan-id-list 700
```

3. Commit the configuration:

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

Results

From operational mode, confirm your configuration by entering the **show configuration interfaces**, **show configuration protocols openflow**, and **show configuration routing-instances** commands. If the output does not display the specified configuration, repeat the configuration instructions in this example to correct the configuration.

```
user@switch> show configuration interfaces
xe-2/1/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members 100-200;
            }
        }
    }
    unit 1 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members 700;
            }
        }
    }
    unit 2 {
        vlan-id 800;
        family inet {
            address 198.51.100.10/24;
        }
    }
}
xe-2/1/1 {
    vlan-tagging;
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members 200;
            }
        }
    }
}
```

```
  xe-2/1/2 {
    vlan-tagging;
    unit 0 {
      family ethernet-switching {
        interface-mode trunk;
        vlan {
          members 700;
        }
      }
    }
  }
```

user@switch> **show configuration protocols openflow**

```
switch 100 {
    interfaces {
        xe-2/1/0.0;
        xe-2/1/1.0;
    }
    controller {
        protocol tcp {
            port 6633;
        }
        address 198.51.100.174;
    }
}
```

user@switch> **show configuration routing-instances**

```
OF {
    instance-type virtual-switch;
    interface xe-2/1/0.0;
    interface xe-2/1/1.0;
    vlans {
        OF-vlan {
            vlan-id-list 100-200;
        }
    }
}
NON-OF {
    instance-type virtual-switch;
    interface xe-2/1/0.1;
```

```
    interface xe-2/1/2.0;
    vlans {
        NOF-vlan {
            vlan-id 700;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the OpenFlow Controller Connection | 146
- Verifying the OpenFlow Interfaces | 147

Confirm that the configuration is working properly.

Verifying the OpenFlow Controller Connection

Purpose

Verify that the OpenFlow controller connection is up.

Action

Issue the **show openflow controller** operational mode command to verify that the controller connection state is **up**. Because the virtual switch configuration has only a single controller, the virtual switch automatically initiates a connection to the controller after you commit the configuration.

`user@switch>` **show openflow controller**

```
Openflowd controller information:
Controller socket: 11
Controller IP address: 198.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
```

```
Number of connection attempt: 5
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying the OpenFlow Interfaces

Purpose

Verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each OpenFlow interface is **Up**.

`user@switch>` **show openflow interfaces**

```
Switch name: 100
Interface Name: xe-2/1/0.0
Interface port number: 41500
Interface Hardware Address: 00:00:5E:00:53:cf
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up

Switch name: 100
Interface Name: xe-2/1/1.0
Interface port number: 41501
Interface Hardware Address: 00:00:5E:00:53:d0
Interface speed: 10Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

RELATED DOCUMENTATION

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS | 119

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches | 134

4

CHAPTER

Configuring OpenFlow Traffic Steering Across MPLS Networks

Understanding OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | **150**

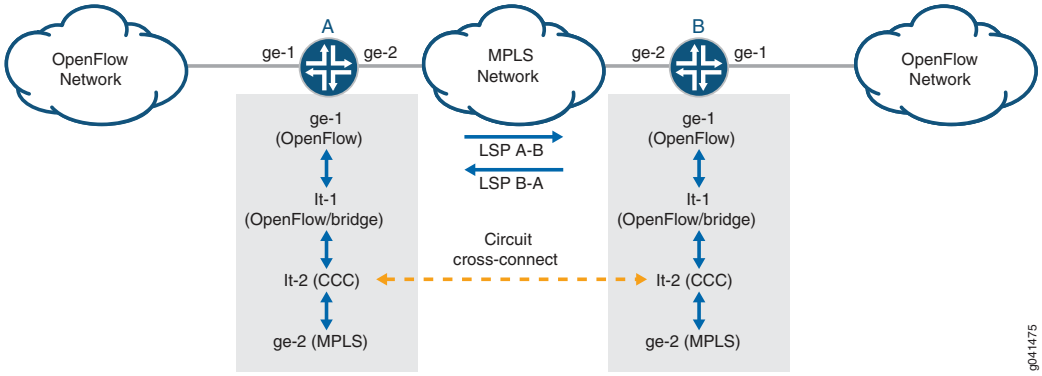
Example: OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | **151**

Understanding OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects

On MX Series devices that support OpenFlow, you can direct traffic from OpenFlow networks over MPLS networks by using logical tunnel interfaces and MPLS LSP tunnel cross-connects. Using logical tunnel interfaces, you can stitch an OpenFlow interface to an MPLS label-switched path (LSP), which enables you to direct traffic from the OpenFlow network onto the MPLS network. MPLS LSP tunnel cross-connects between interfaces and LSPs permit you to connect the OpenFlow network to a remote network by creating MPLS tunnels that use LSPs as the conduit.

The topology in Figure 1 on page 150 illustrates an MPLS LSP tunnel cross-connect that connects two remote OpenFlow networks through an MPLS network. Circuit cross-connect (CCC) enables you to establish an LSP tunnel between the two domains, through which you can tunnel the traffic from one OpenFlow network across the MPLS network to the second OpenFlow network.

Figure 1: Connecting OpenFlow Networks Using MPLS LSP Tunnel Cross-Connects



Router A and Router B are OpenFlow-enabled routers that have MPLS LSPs configured to route traffic across the MPLS network. LSP A-B routes traffic from Router A to Router B, and LSP B-A routes traffic from Router B to Router A.

Each router has an OpenFlow interface, ge-1, and an MPLS interface, ge-2. You can stitch the OpenFlow interface to the MPLS LSP by using two logical tunnel interfaces. You configure the first logical tunnel interface, lt-1, as a Layer 2 interface that participates in OpenFlow. The second logical tunnel interface, lt-2, uses CCC encapsulation. You configure lt-1 and lt-2 interfaces as peers, so that traffic entering one logical interface is automatically directed to the second logical interface.

On each router, MPLS LSP tunnel cross-connects are configured at the **[edit protocols connections remote-interface-switch]** hierarchy level. The cross-connects make an association between the CCC

interface, lt-2, and the two LSPs, one for transmitting MPLS packets from the local device to the remote device and the other for receiving MPLS packets on the local device from the remote device.

For traffic flowing from Router A to Router B, the OpenFlow controller must install flow entries on Router A that direct the desired OpenFlow traffic from ge-1 as the OpenFlow ingress port to lt-1 as the output port. On Router B, the OpenFlow controller must install flow entries that direct the OpenFlow traffic from lt-1 as the OpenFlow ingress port to ge-1 as the output port. Similarly for traffic flowing from Router B to Router A, the OpenFlow controller must install flow entries on Router B that direct the desired OpenFlow traffic from ge-1 as the OpenFlow ingress port to lt-1 as the output port. On Router A, the OpenFlow controller must install flow entries that direct the OpenFlow traffic from lt-1 as the OpenFlow ingress port to ge-1 as the output port.

RELATED DOCUMENTATION

Example: OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | 151

Understanding Support for OpenFlow on Devices Running Junos OS | 18

Example: OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects

IN THIS SECTION

- Requirements | 152
- Overview | 152
- Configuration | 153
- Verification | 170
- Troubleshooting | 174

On MX Series routers that support OpenFlow, you can direct traffic from OpenFlow networks over MPLS networks by using logical tunnel interfaces and MPLS LSP tunnel cross-connects. This example shows how to configure MX Series routers to send traffic between two remote OpenFlow networks over an MPLS-based network using MPLS LSP tunnel cross-connects.

Requirements

This example uses the following hardware and software components for the OpenFlow-enabled routers:

- MX240 routers running Junos OS Release 13.3 or a later release.
- OpenFlow software package with a software package release that matches the Junos OS release of the device on which it is installed
- TCP connection between the router and an OpenFlow controller
- Connection between the fxp0 management interface of the router and the management network, which is reachable from the controller IP address

Overview

In this example, you configure MPLS LSP tunnel cross-connects to connect two remote OpenFlow networks that are separated by an MPLS network. Figure 2 on page 153 shows the topology used in this example.

This example has three routers: a provider router (P) and two provider edge routers (PE1 and PE2). Router P resides within an MPLS network. Routers PE1 and PE2 are OpenFlow-enabled routers, each with the ge-1/0/0.0 interface configured to accept and forward OpenFlow traffic and two MPLS interfaces that connect to Router P. The network uses OSPF as the IPG, and it has two LSPs: LSP 1-3 routes traffic from PE1 to PE2, and LSP 3-1 routes traffic from PE2 to PE1.

You stitch the OpenFlow interface to the MPLS LSP using two logical tunnel interfaces, lt-1/1/10.0 and lt-1/1/10.100. You configure the first logical tunnel interface, lt-1/1/10.0, as a Layer 2 interface with encapsulation **ethernet-bridge** and family **bridge**. This interface participates in OpenFlow. The second logical tunnel interface, lt-1/1/10.100, uses circuit cross-connect (CCC) encapsulation. You configure lt-1 and lt-2 interfaces as peers, so that traffic entering one logical interface is automatically directed to the second logical interface.

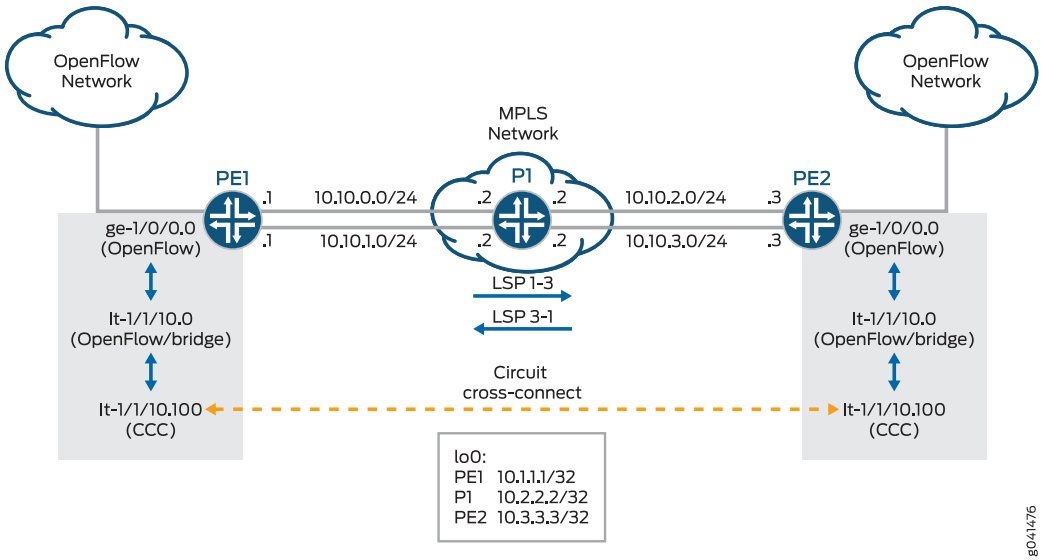
On the PE1 and PE2 routers, you configure an MPLS LSP tunnel cross-connect at the **[edit protocols connections remote-interface-switch]** hierarchy level using the logical tunnel interface with CCC encapsulation. This configuration makes an association between the CCC interface and two LSPs, one for transmitting MPLS packets from the local device to the remote device and the other for receiving MPLS packets on the local device from the remote device.

For traffic flowing from PE1 to PE2, the OpenFlow controller must install flow entries on PE1 that direct the desired OpenFlow traffic from ge-1/0/0.0 as the OpenFlow ingress port to lt-1/1/10.0 as the output port. On PE2, the OpenFlow controller must install flow entries that direct the OpenFlow traffic from lt-1/1/10.0 as the OpenFlow ingress port to ge-1/0/0.0 as the output port. Similarly, for traffic flowing from PE2 to PE1, the OpenFlow controller must install flow entries on PE2 that direct the desired OpenFlow traffic from ge-1/0/0.0 as the OpenFlow ingress port to lt-1/1/10.0 as the output port. On PE1, the

OpenFlow controller must install flow entries that direct the OpenFlow traffic from lt-1/1/10.0 as the OpenFlow ingress port to ge-1/1/0.0 as the output port.

Topology

Figure 2: Connecting OpenFlow Networks Using MPLS Tunnel Cross-Connects



Configuration

IN THIS SECTION

- Configuring the Ingress Provider Edge Router (PE1) | **156**
- Configuring the Provider Router (P) | **161**
- Configuring the Egress Provider Edge Router (PE2) | **165**

CLI Quick Configuration

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

Device PE1

```
set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
set interfaces ge-1/0/0 encapsulation ethernet-bridge
set interfaces ge-1/0/0 unit 0 family bridge
set interfaces ge-1/1/0 unit 0 family inet address 10.10.0.1/24
set interfaces ge-1/1/0 unit 0 family mpls
set interfaces ge-1/1/1 unit 0 family inet address 10.10.1.1/24
set interfaces ge-1/1/1 unit 0 family mpls
set interfaces lt-1/1/10 unit 0 encapsulation ethernet-bridge
set interfaces lt-1/1/10 unit 0 peer-unit 100
set interfaces lt-1/1/10 unit 0 family bridge
set interfaces lt-1/1/10 unit 100 encapsulation ethernet-ccc
set interfaces lt-1/1/10 unit 100 peer-unit 0
set interfaces lt-1/1/10 unit 100 family ccc
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set protocols rsvp interface ge-1/1/0.0
set protocols rsvp interface ge-1/1/1.0
set protocols mpls label-switched-path 1-3 from 10.1.1.1
set protocols mpls label-switched-path 1-3 to 10.3.3.3
set protocols mpls interface ge-1/1/0.0
set protocols mpls interface ge-1/1/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-1/1/0.0
set protocols ospf area 0.0.0.0 interface ge-1/1/1.0
set protocols connections remote-interface-switch 1-3-ccc interface lt-1/1/10.100
set protocols connections remote-interface-switch 1-3-ccc transmit-lsp 1-3
set protocols connections remote-interface-switch 1-3-ccc receive-lsp 3-1
set protocols openflow switch s1 interfaces ge-1/0/0.0 port-id 1
set protocols openflow switch s1 interfaces lt-1/1/10.0 port-id 2
set protocols openflow switch s1 controller protocol tcp port 6633
set protocols openflow switch s1 controller address 10.94.175.213
set routing-instances r1 instance-type virtual-switch
set routing-instances r1 bridge-domains bd1 interface ge-1/0/0.0
set routing-instances r1 bridge-domains bd1 interface lt-1/1/10.0
set routing-options router-id 10.1.1.1
```

Device P

```
set interfaces ge-1/1/0 unit 0 family inet address 10.10.0.2/24
set interfaces ge-1/1/0 unit 0 family mpls
set interfaces ge-1/1/1 unit 0 family inet address 10.10.1.2/24
```

```
set interfaces ge-1/1/1 unit 0 family mpls
set interfaces ge-1/1/2 unit 0 family inet address 10.10.2.2/24
set interfaces ge-1/1/2 unit 0 family mpls
set interfaces ge-1/1/3 unit 0 family inet address 10.10.3.2/24
set interfaces ge-1/1/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.2.2.2/32
set protocols rsvp interface ge-1/1/0.0
set protocols rsvp interface ge-1/1/1.0
set protocols rsvp interface ge-1/1/2.0
set protocols rsvp interface ge-1/1/3.0
set protocols mpls interface ge-1/1/0.0
set protocols mpls interface ge-1/1/1.0
set protocols mpls interface ge-1/1/2.0
set protocols mpls interface ge-1/1/3.0
set protocols mpls interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-1/1/0.0
set protocols ospf area 0.0.0.0 interface ge-1/1/1.0
set protocols ospf area 0.0.0.0 interface ge-1/1/2.0
set protocols ospf area 0.0.0.0 interface ge-1/1/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 10.2.2.2
```

Device PE2

```
set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
set interfaces ge-1/0/0 encapsulation ethernet-bridge
set interfaces ge-1/0/0 unit 0 family bridge
set interfaces ge-1/1/2 unit 0 family inet address 10.10.2.3/24
set interfaces ge-1/1/2 unit 0 family mpls
set interfaces ge-1/1/3 unit 0 family inet address 10.10.3.3/24
set interfaces ge-1/1/3 unit 0 family mpls
set interfaces lt-1/1/10 unit 0 encapsulation ethernet-bridge
set interfaces lt-1/1/10 unit 0 peer-unit 100
set interfaces lt-1/1/10 unit 0 family bridge
set interfaces lt-1/1/10 unit 100 encapsulation ethernet-ccc
set interfaces lt-1/1/10 unit 100 peer-unit 0
set interfaces lt-1/1/10 unit 100 family ccc
set interfaces lo0 unit 0 family inet address 10.3.3.3/32
```

```
set protocols rsvp interface ge-1/1/2.0
set protocols rsvp interface ge-1/1/3.0
set protocols mpls label-switched-path 3-1 from 10.3.3.3
set protocols mpls label-switched-path 3-1 to 10.1.1.1
set protocols mpls interface ge-1/1/2.0
set protocols mpls interface ge-1/1/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-1/1/2.0
set protocols ospf area 0.0.0.0 interface ge-1/1/3.0
set protocols connections remote-interface-switch 3-1-ccc interface lt-1/1/10.100
set protocols connections remote-interface-switch 3-1-ccc transmit-lsp 3-1
set protocols connections remote-interface-switch 3-1-ccc receive-lsp 1-3
set protocols openflow switch s1 interfaces ge-1/0/0.0 port-id 1
set protocols openflow switch s1 interfaces lt-1/1/10.0 port-id 2
set protocols openflow switch s1 controller protocol tcp port 6633
set protocols openflow switch s1 controller address 10.94.175.213
set routing-instances r1 instance-type virtual-switch
set routing-instances r1 bridge-domains bd1 interface ge-1/0/0.0
set routing-instances r1 bridge-domains bd1 interface lt-1/1/10.0
set routing-options router-id 10.3.3.3
```

Configuring the Ingress Provider Edge Router (PE1)

Step-by-Step Procedure

To configure Router PE1:

1. Create tunnel interfaces by configuring the DPC and its corresponding PIC to use tunneling services.

```
[edit]
user@PE1# set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
```

2. Configure the OpenFlow interface as a Layer 2 interface.

```
[edit interfaces]
user@PE1# set ge-1/0/0 encapsulation ethernet-bridge
user@PE1# set ge-1/0/0 unit 0 family bridge
```

3. Configure the OpenFlow virtual switch routing instance.

```
[edit]
user@PE1# set routing-instances r1 instance-type virtual-switch
user@PE1# set routing-instances r1 bridge-domains bd1 interface ge-1/0/0.0
user@PE1# set routing-instances r1 bridge-domains bd1 interface lt-1/1/10.0
```

4. Configure the OpenFlow controller.

```
[edit protocols openflow]
user@PE1# set switch s1 controller address 10.94.175.213
user@PE1# set switch s1 controller protocol tcp port 6633
```

5. Configure the interfaces participating in OpenFlow.

```
[edit protocols openflow]
user@PE1# set switch s1 interfaces ge-1/0/0.0 port-id 1
user@PE1# set switch s1 interfaces lt-1/1/10.0 port-id 2
```

6. Configure the loopback interface and router ID.

```
[edit]
user@PE1# set interfaces lo0 unit 0 family inet address 10.1.1.1/32
user@PE1# set routing-options router-id 10.1.1.1
```

7. Configure the MPLS interfaces.

```
[edit interfaces]
user@PE1# set ge-1/1/0 unit 0 family inet address 10.10.0.1/24
user@PE1# set ge-1/1/0 unit 0 family mpls
user@PE1# set ge-1/1/1 unit 0 family inet address 10.10.1.1/24
user@PE1# set ge-1/1/1 unit 0 family mpls
```

8. Configure the logical tunnel interface.

```
[edit interfaces]
user@PE1# set lt-1/1/10 unit 0 family bridge
user@PE1# set lt-1/1/10 unit 0 encapsulation ethernet-bridge
```

```
user@PE1# set lt-1/1/10 unit 0 peer-unit 100
user@PE1# set lt-1/1/10 unit 100 family ccc
user@PE1# set lt-1/1/10 unit 100 encapsulation ethernet-ccc
user@PE1# set lt-1/1/10 unit 100 peer-unit 0
```

9. Enable RSVP, MPLS, and OSPF on the interfaces connected to Router P.

```
[edit protocols]
user@PE1# set rsvp interface ge-1/1/0.0
user@PE1# set rsvp interface ge-1/1/1.0
user@PE1# set mpls interface ge-1/1/0.0
user@PE1# set mpls interface ge-1/1/1.0
user@PE1# set ospf area 0.0.0.0 interface ge-1/1/0.0
user@PE1# set ospf area 0.0.0.0 interface ge-1/1/1.0
```

10. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
```

11. Configure the MPLS LSP from PE1 to PE2.

```
[edit protocols]
user@PE1# set mpls label-switched-path 1-3 from 10.1.1.1
user@PE1# set mpls label-switched-path 1-3 to 10.3.3.3
```

12. Configure the MPLS LSP tunnel cross-connects.

```
[edit protocols]
user@PE1# set connections remote-interface-switch 1-3-ccc interface lt-1/1/10.100
user@PE1# set connections remote-interface-switch 1-3-ccc transmit-lsp 1-3
user@PE1# set connections remote-interface-switch 1-3-ccc receive-lsp 3-1
```

13. Commit the configuration.

```
[edit]
user@PE1# commit
```

Results

From configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
chassis {
    fpc 1 {
        pic 1 {
            tunnel-services {
                bandwidth 1g;
            }
        }
    }
}

interfaces {
    ge-1/0/0 {
        encapsulation ethernet-bridge;
        unit 0 {
            family bridge;
        }
    }
    ge-1/1/0 {
        unit 0 {
            family inet {
                address 10.10.0.1/24;
            }
            family mpls;
        }
    }
    ge-1/1/1 {
        unit 0 {
            family inet {
                address 10.10.1.1/24;
            }
            family mpls;
        }
    }
    lt-1/1/10 {
        unit 0 {
            encapsulation ethernet-bridge;
            peer-unit 100;
            family bridge;
```

```
        }
        unit 100 {
            encapsulation ethernet-ccc;
            peer-unit 0;
            family ccc;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.1.1/32;
            }
        }
    }
}

protocols {
    rsvp {
        interface ge-1/1/0.0;
        interface ge-1/1/1.0;
    }
    mpls {
        label-switched-path 1-3 {
            from 10.1.1.1;
            to 10.3.3.3;
        }
        interface ge-1/1/0.0;
        interface ge-1/1/1.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-1/1/0.0;
            interface ge-1/1/1.0;
        }
    }
    connections {
        remote-interface-switch 1-3-ccc {
            interface lt-1/1/10.100;
            transmit-lsp 1-3;
            receive-lsp 3-1;
        }
    }
    openflow {
```

```
        switch s1 {
            interfaces {
                ge-1/0/0.0 port-id 1;
                lt-1/1/10.0 port-id 2;
            }
            controller {
                protocol {
                    tcp {
                        port 6633;
                    }
                }
                address 10.94.175.213;
            }
        }
    }
}

routing-instances {
    r1 {
        instance-type virtual-switch;
        bridge-domains {
            bd1 {
                interface ge-1/0/0.0;
                interface lt-1/1/10.0;
            }
        }
    }
}

routing-options {
    router-id 10.1.1.1;
}
...
```

Configuring the Provider Router (P)

Step-by-Step Procedure

To configure Router P:

1. Configure the loopback interface and router ID.

```
[edit]
user@P# set interfaces lo0 unit 0 family inet address 10.2.2.2/32
```

```
user@P# set routing-options router-id 10.2.2.2
```

2. Configure the MPLS interfaces.

```
[edit interfaces]
user@P# set ge-1/1/0 unit 0 family inet address 10.10.0.2/24
user@P# set ge-1/1/0 unit 0 family mpls
user@P# set ge-1/1/1 unit 0 family inet address 10.10.1.2/24
user@P# set ge-1/1/1 unit 0 family mpls
user@P# set ge-1/1/2 unit 0 family inet address 10.10.2.2/24
user@P# set ge-1/1/2 unit 0 family mpls
user@P# set ge-1/1/3 unit 0 family inet address 10.10.3.2/24
user@P# set ge-1/1/3 unit 0 family mpls
```

3. Enable RSVP, MPLS, and OSPF on the interfaces connected to PE1 and PE2.

```
[edit protocols]
user@P# set rsvp interface ge-1/1/0.0
user@P# set rsvp interface ge-1/1/1.0
user@P# set rsvp interface ge-1/1/2.0
user@P# set rsvp interface ge-1/1/3.0
user@P# set mpls interface lo0.0
user@P# set mpls interface ge-1/1/0.0
user@P# set mpls interface ge-1/1/1.0
user@P# set mpls interface ge-1/1/2.0
user@P# set mpls interface ge-1/1/3.0
user@P# set ospf area 0.0.0.0 interface fxp0.0 disable
user@P# set ospf area 0.0.0.0 interface ge-1/1/0.0
user@P# set ospf area 0.0.0.0 interface ge-1/1/1.0
user@P# set ospf area 0.0.0.0 interface ge-1/1/2.0
user@P# set ospf area 0.0.0.0 interface ge-1/1/3.0
user@P# set ospf area 0.0.0.0 interface lo0.0
```

4. Enable traffic engineering for OSPF.

```
[edit protocols]
user@P# set ospf traffic-engineering
```

5. Commit the configuration.

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

Results

From configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
interfaces {
    ge-1/1/0 {
        unit 0 {
            family inet {
                address 10.10.0.2/24;
            }
            family mpls;
        }
    }
    ge-1/1/1 {
        unit 0 {
            family inet {
                address 10.10.1.2/24;
            }
            family mpls;
        }
    }
    ge-1/1/2 {
        unit 0 {
            family inet {
                address 10.10.2.2/24;
            }
            family mpls;
        }
    }
    ge-1/1/3 {
        unit 0 {
            family inet {
                address 10.10.3.2/24;
            }
            family mpls;
        }
```

```
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.2.2.2/32;
            }
        }
    }
}

protocols {
    rsvp {
        interface ge-1/1/0.0;
        interface ge-1/1/1.0;
        interface ge-1/1/2.0;
        interface ge-1/1/3.0;
    }
    mpls {
        interface ge-1/1/0.0;
        interface ge-1/1/1.0;
        interface ge-1/1/2.0;
        interface ge-1/1/3.0;
        interface lo0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface ge-1/1/0.0;
            interface ge-1/1/1.0;
            interface ge-1/1/2.0;
            interface ge-1/1/3.0;
            interface lo0.0;
        }
    }
}

routing-options {
    router-id 10.2.2.2;
}
...
```

Configuring the Egress Provider Edge Router (PE2)

Step-by-Step Procedure

To configure Router PE2:

1. Create tunnel interfaces by configuring the DPC and its corresponding PIC to use tunneling services.

```
[edit]
user@PE2# set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
```

2. Configure the OpenFlow interface as a Layer 2 interface.

```
[edit interfaces]
user@PE2# set ge-1/0/0 encapsulation ethernet-bridge
user@PE2# set ge-1/0/0 unit 0 family bridge
```

3. Configure the OpenFlow virtual switch routing instance.

```
[edit]
user@PE2# set routing-instances r1 instance-type virtual-switch
user@PE2# set routing-instances r1 bridge-domains bd1 interface ge-1/0/0.0
user@PE2# set routing-instances r1 bridge-domains bd1 interface lt-1/1/10.0
```

4. Configure the OpenFlow controller.

```
[edit protocols openflow]
user@PE2# set switch s1 controller protocol tcp port 6633
user@PE2# set switch s1 controller address 10.94.175.213
```

5. Configure the interfaces participating in OpenFlow.

```
[edit protocols openflow]
user@PE2# set switch s1 interfaces ge-1/0/0.0 port-id 1
user@PE2# set switch s1 interfaces lt-1/1/10.0 port-id 2
```

6. Configure the loopback interface and router ID.

```
[edit]
```

```
user@PE2# set interfaces lo0 unit 0 family inet address 10.3.3.3/32
user@PE2# set routing-options router-id 10.3.3.3
```

7. Configure the MPLS interfaces.

```
[edit interfaces]
user@PE2# set ge-1/1/2 unit 0 family inet address 10.10.2.3/24
user@PE2# set ge-1/1/2 unit 0 family mpls
user@PE2# set ge-1/1/3 unit 0 family inet address 10.10.3.3/24
user@PE2# set ge-1/1/3 unit 0 family mpls
```

8. Configure the logical tunnel interface.

```
[edit interfaces]
user@PE2# set lt-1/1/10 unit 0 family bridge
user@PE2# set lt-1/1/10 unit 0 encapsulation ethernet-bridge
user@PE2# set lt-1/1/10 unit 0 peer-unit 100
user@PE2# set lt-1/1/10 unit 100 family ccc
user@PE2# set lt-1/1/10 unit 100 encapsulation ethernet-ccc
user@PE2# set lt-1/1/10 unit 100 peer-unit 0
```

9. Enable RSVP, MPLS, and OSPF on the interfaces connected to Router P.

```
[edit protocols]
user@PE2# set rsvp interface ge-1/1/2.0
user@PE2# set rsvp interface ge-1/1/3.0
user@PE2# set mpls interface ge-1/1/2.0
user@PE2# set mpls interface ge-1/1/3.0
user@PE2# set ospf area 0.0.0.0 interface ge-1/1/2.0
user@PE2# set ospf area 0.0.0.0 interface ge-1/1/3.0
```

10. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PE2# set ospf traffic-engineering
```

11. Configure the MPLS LSP from PE2 to PE1.

```
[edit protocols]
user@PE2# set mpls label-switched-path 3-1 from 10.3.3.3
user@PE2# set mpls label-switched-path 3-1 to 10.1.1.1
```

12. Configure the MPLS LSP tunnel cross-connects.

```
[edit protocols]
user@PE2# set connections remote-interface-switch 3-1-ccc interface lt-1/1/10.100
user@PE2# set connections remote-interface-switch 3-1-ccc transmit-lsp 3-1
user@PE2# set connections remote-interface-switch 3-1-ccc receive-lsp 1-3
```

13. Commit the configuration.

```
[edit]
user@PE2# commit
```

Results

From configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
chassis {
    fpc 1 {
        pic 1 {
            tunnel-services {
                bandwidth 1g;
            }
        }
    }
}

interfaces {
    ge-1/0/0 {
        encapsulation ethernet-bridge;
        unit 0 {
            family bridge;
        }
```

```
    }
    ge-1/1/2 {
        unit 0 {
            family inet {
                address 10.10.2.3/24;
            }
            family mpls;
        }
    }
    ge-1/1/3 {
        unit 0 {
            family inet {
                address 10.10.3.3/24;
            }
            family mpls;
        }
    }
    lt-1/1/10 {
        unit 0 {
            encapsulation ethernet-bridge;
            peer-unit 100;
            family bridge;
        }
        unit 100 {
            encapsulation ethernet-ccc;
            peer-unit 0;
            family ccc;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.3.3.3/32;
            }
        }
    }
}

protocols {
    rsvp {
        interface ge-1/1/2.0;
        interface ge-1/1/3.0;
    }
    mpls {
```

```
        label-switched-path 3-1 {
            from 10.3.3.3;
            to 10.1.1.1;
        }
        interface ge-1/1/2.0;
        interface ge-1/1/3.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-1/1/2.0;
            interface ge-1/1/3.0;
        }
    }
    connections {
        remote-interface-switch 3-1-ccc {
            interface lt-1/1/10.100;
            transmit-lsp 3-1;
            receive-lsp 1-3;
        }
    }
    openflow {
        switch s1 {
            interfaces {
                ge-1/0/0.0 port-id 1;
                lt-1/1/10.0 port-id 2;
            }
            controller {
                protocol {
                    tcp {
                        port 6633;
                    }
                }
                address 10.94.175.213;
            }
        }
    }
}


routing-instances {
    r1 {
        instance-type virtual-switch;
        bridge-domains {
```

```
            bd1 {
                interface ge-1/0/0.0;
                interface lt-1/1/10.0;
            }
        }
    }
}

routing-options {
    router-id 10.3.3.3;
}
...
```

Verification

IN THIS SECTION

- Verifying that the OpenFlow Controller Connection Is Up | 170
- Verifying that the OpenFlow Interfaces Are Up | 171
- Verifying that the MPLS LSP Is Operational | 172
- Verifying that the MPLS LSP Cross-Connect Is Operational | 173
- Verifying the Routes | 174

Confirm that the configuration is working properly.

Verifying that the OpenFlow Controller Connection Is Up

Purpose

On each of the OpenFlow-enabled routers, verify that the connection state for the OpenFlow controller is **up**.

Action

Issue the **show openflow controller** operational mode command, and verify that the controller connection state is **up**.

```
user@PE1> show openflow controller
```

```
Openflowd controller information:
Controller socket: 11
Controller IP address: 10.94.175.213
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 1
Controller role: equal
```

Meaning

The output shows that the connection state of the OpenFlow controller is **up**, in addition to other information about the controller.

Verifying that the OpenFlow Interfaces Are Up

Purpose

On each of the OpenFlow-enabled routers, verify that the OpenFlow interfaces are up.

Action

Issue the **show openflow interfaces** operational mode command, and verify that the state of each interface is **Up**. For example, on PE1:

`user@PE1>` **show openflow interfaces**

```
Switch name: s1
Interface Name: ge-1/0/0.0
Interface port number: 1
Interface Hardware Address: 00:00:5e:00:53:b1
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: s1
Interface Name: lt-1/1/10.0
Interface port number: 2
Interface Hardware Address: 00:00:5e:00:53:be
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Disabled
Interface media type: Fiber
Interface state: Up
```

Meaning

The output shows that the state of each OpenFlow interface is **Up**, in addition to other information about the interfaces.

Verifying that the MPLS LSP Is Operational

Purpose

On each edge router, verify that the MPLS LSP state is **Up**.

Action

Issue the **show mpls lsp** operational mode command, and verify that each LSP is operational.

```
user@PE1> show mpls lsp
```

```
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.3.3.3        10.1.1.1        Up     0 *                      1-3
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.1.1.1        10.3.3.3        Up       0  1 FF  299776        - 3-1
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

```
user@PE2> show mpls lsp
```

```
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.1.1.1        10.3.3.3        Up     0 *                      3-1
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.3.3.3        10.1.1.1        Up       0  1 FF  299840        - 1-3
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The output shows that each LSP is operational.

Verifying that the MPLS LSP Cross-Connect Is Operational

Purpose

Verify that the MPLS LSP circuit cross-connect is operational.

Action

Issue the **show connections remote-interface-switch** operational mode command, and verify that the circuit cross-connect state is **Up**.

```
user@PE1> show connections remote-interface-switch
```

```
CCC and TCC connections [Link Monitoring On]
[...Output truncated...]

Connection/Circuit               Type        St      Time last up     # Up trans
1-3-ccc                          rmt-if      Up      Apr 18 22:30:54           1

  lt-1/1/10.100                    intf  Up
  1-3                              tlsp  Up
  3-1                              rlsp  Up
```

```
user@PE2> show connections remote-interface-switch
```

```
CCC and TCC connections [Link Monitoring On]
[...Output truncated...]

Connection/Circuit               Type        St      Time last up     # Up trans
3-1-ccc                          rmt-if      Up      Apr 18 15:07:04           1

  lt-1/1/10.100                    intf  Up
  3-1                              tlsp  Up
  1-3                              rlsp  Up
```

Meaning

The output from both routers indicates that the circuit cross-connect is operational.

Verifying the Routes

Purpose

Ensure that the routes from the CCC interface over the LSP are active.

Action

Issue the **show route ccc lt-1/1/10.100** command.

`user@PE1>` **show route ccc lt-1/1/10.100**

```
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

lt-1/1/10.100      *[CCC/7/1] 00:34:54, metric 2
                    > to 10.10.1.2 via ge-1/1/1.0, label-switched-path 1-3
```

`user@PE2>` **show route ccc lt-1/1/10.100**

```
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

lt-1/1/10.100      *[CCC/7/1] 00:35:48, metric 2
                    > to 10.10.2.2 via ge-1/1/2.0, label-switched-path 3-1
```

Meaning

The sample output shows that the circuit cross-connect uses the configured LSPs with the MPLS interface as the exit interface.

Troubleshooting

Troubleshooting the Circuit Cross-Connect

Problem

The OpenFlow-enabled router does not route OpenFlow traffic to the remote OpenFlow network.

Solution

In order to direct traffic from the local OpenFlow network to the remote OpenFlow network, the OpenFlow controller must install flow entries that select the appropriate traffic and forward it to the correct OpenFlow interface. For traffic flowing from PE1 to PE2, the OpenFlow controller must install flow entries on PE1 that direct OpenFlow traffic from ge-1/0/0.0 to lt-1/1/10.0, and it must install flow entries on PE2 that direct the OpenFlow traffic from lt-1/1/10.0 to ge-1/0/0.0. Similarly, for traffic flowing from PE2 to PE1, the OpenFlow controller must install flow entries on PE2 that direct the desired OpenFlow traffic from ge-1/0/0.0 to lt-1/1/10.0, and it must install flow entries on PE1 that direct the OpenFlow traffic from lt-1/1/10.0 to ge-1/1/0.0.

RELATED DOCUMENTATION

Understanding OpenFlow Traffic Steering Across MPLS Networks Using MPLS LSP Tunnel Cross-Connects | **150**

Configuring Support for OpenFlow on MX Series Routers | **78**

5

CHAPTER

Configuration Statements

address (Protocols OpenFlow) | **177**

controller (Protocols OpenFlow) | **178**

default-action (Protocols OpenFlow) | **180**

id (Protocols OpenFlow) | **181**

interfaces (Protocols OpenFlow) | **182**

openflow (Protocols OpenFlow) | **183**

port (Protocols OpenFlow) | **185**

protocol (Protocols OpenFlow) | **186**

purge-flow-timer (Protocols OpenFlow) | **187**

role (Protocols OpenFlow) | **188**

switch (Protocols OpenFlow) | **189**

traceoptions (Protocols OpenFlow) | **191**

address (Protocols OpenFlow)

Syntax

```
address address;
```

Hierarchy Level

```
[edit protocols openflow switch switch-name controller]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify the IPv4 address of the OpenFlow controller that will manage OpenFlow on that virtual switch. The switch establishes a connection to the controller using this address.

Options

address—IPv4 address of the OpenFlow controller.

Required Privilege Level

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

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- controller (Protocols OpenFlow) | 178
- protocol (Protocols OpenFlow) | 186
- switch (Protocols OpenFlow) | 189

controller (Protocols OpenFlow)

Syntax

```
controller {
    address address;
    id id;
    protocol tcp {
        port port;
    }
    role equal;
}
```

Hierarchy Level

[edit protocols **openflow switch** *switch-name*]

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Configure the OpenFlow controller connection information for a virtual switch on an OpenFlow-enabled device running Junos OS. If you configure a virtual switch with a single controller, by default, the controller is in active mode, and the switch automatically initiates a connection to the controller.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS | **26**
- address (Protocols OpenFlow) | **177**

protocol (Protocols OpenFlow) | 186

role (Protocols OpenFlow) | 188

switch (Protocols OpenFlow) | 189

default-action (Protocols OpenFlow)

Syntax

```
default-action (drop | packet-in);
```

Hierarchy Level

```
[edit protocols openflow switch switch-name]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify the default action that is executed when an OpenFlow packet does not match an existing flow entry. The default action is specific to the OpenFlow virtual switch and is the same across all filters associated with that virtual switch.

Default

If you do not include the **default-action** statement, the default action is **packet-in**.

Options

drop—Drop packets that do not match an existing flow entry.

packet-in—Accept packets that do not match an existing flow entry, and forward the packet to the controller.

Required Privilege Level

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

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- Understanding OpenFlow Flows and Filters on Devices Running Junos OS | 29
- openflow (Protocols OpenFlow) | 183
- switch (Protocols OpenFlow) | 189

id (Protocols OpenFlow)

Syntax

```
id id;
```

Hierarchy Level

```
[edit protocols openflow switch switch-name controller]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify an optional numeric identifier for the OpenFlow controller.

Options

id—Numeric identifier for the OpenFlow controller.

Required Privilege Level

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

controller (Protocols OpenFlow) | 178

interfaces (Protocols OpenFlow)

Syntax

```
interfaces {
    interface-name port-id port;
}
```

Hierarchy Level

[edit protocols **openflow switch** *switch-name*]

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Configure a Layer 2 interface as an OpenFlow-enabled interface.

Options

interface-name—Name of the interface, including the logical unit number—for example, ge-1/1/0.0.

port-id *port*—(Optional) Unique numeric value specifying the port ID associated with the OpenFlow interface. You can manually configure a port ID in the range 1 through 32640. If you do not specify a port, the system generates a value in the range from 32641 through 65280.

Range: 1 through 32,640

Required Privilege Level

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | **18**

openflow (Protocols OpenFlow) | **183**

switch (Protocols OpenFlow) | **189**

openflow (Protocols OpenFlow)

Syntax

```
openflow {
    switch switch-name {
        controller {
            address address;
            id id;
            protocol tcp {
                port port;
            }
            role equal;
        }
        default-action (drop | packet-in);
        interfaces {
            interface-name port-id port;
        }
        purge-flow-timer seconds;
    }
    traceoptions {
        file <filename> <files number> <match regular-expression> <size size> <world-readable | no-world-readable>;
        flag flag;
    }
}
```

Hierarchy Level

[edit protocols]

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Configure support for OpenFlow on a device running Junos OS. To configure OpenFlow, the device must be running a Junos OS release that supports OpenFlow and have the OpenFlow software package installed. The OpenFlow software package release must match the Junos OS release of the device on which the software is installed.

Default

OpenFlow is disabled on the device.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

- OpenFlow Support on Juniper Networks Devices | **17**
- Understanding Support for OpenFlow on Devices Running Junos OS | **18**
- OpenFlow Operational Mode Commands | **196**

port (Protocols OpenFlow)

Syntax

```
port port;
```

Hierarchy Level

```
[edit protocols openflow switch switch-name controller protocol protocol]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify the OpenFlow controller port to which the OpenFlow virtual switch connects.

Options

port—Numeric value specifying the OpenFlow controller port to which the device should connect.

Range: 1024 through 65,535

Default: 6633

Required Privilege Level

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

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- address (Protocols OpenFlow) | 177
- controller (Protocols OpenFlow) | 178
- protocol (Protocols OpenFlow) | 186

protocol (Protocols OpenFlow)

Syntax

```
protocol tcp {
    port port;
}
```

Hierarchy Level

```
[edit protocols openflow switch switch-name controller]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify the connection protocol that the OpenFlow virtual switch uses to connect to the OpenFlow controller.

Options

tcp—Establish a TCP connection to the controller.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

- Understanding Support for OpenFlow on Devices Running Junos OS | 18
- controller (Protocols OpenFlow) | 178
- port (Protocols OpenFlow) | 185

purge-flow-timer (Protocols OpenFlow)

Syntax

```
purge-flow-timer seconds;
```

Hierarchy Level

```
[edit protocols openflow switch switch-name]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

For an OpenFlow virtual switch, specify the number of seconds after which an invalid OpenFlow flow entry is deleted from the flow table.

If you do not configure the **purge-flow-timer** statement, the device purges invalid flows from hardware, but indefinitely retains the corresponding flow entries in the flow table. If you configure the **purge-flow-timer** statement, the device purges invalid flows from hardware, and after the specified number of seconds, the device deletes the invalid flow entries from the flow table. Configuring a value of 0 causes the device to immediately delete invalid flow entries from the flow table.

> **NOTE:** By default, if you remove an active OpenFlow interface from an existing OpenFlow configuration, flow entries that match on this interface as the ingress interface and flow entries that include this interface in their action list (for OpenFlow v1.0) or flow instructions (for OpenFlow v1.3.1) are invalid and are automatically purged from the flow table and from the hardware regardless of whether you configure the **purge-flow-timer** statement.

Options

seconds—Number of seconds after which an invalid flow entry is deleted from the flow table.

Range: 0 through 300

Required Privilege Level

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18
Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS | 34

role (Protocols OpenFlow)

Syntax

```
role equal;
```

Hierarchy Level

[edit protocols **openflow switch** *switch-name* **controller**]

Release Information
Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Specify the role of each OpenFlow controller when configuring more than one controller for a virtual switch. A single controller configuration automatically puts the controller in active mode. In active mode, the virtual switch automatically initiates a connection to the controller.

Options

equal—Configure the controller as the active controller in a single controller configuration.

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18
controller (Protocols OpenFlow) | 178

switch (Protocols OpenFlow)

Syntax

```
switch switch-name {
    controller {
        address address;
        id id;
        protocol tcp {
            port port;
        }
        role equal;
    }
    default-action (drop | packet-in);
    interfaces {
        interface-name port-id port;
    }
    purge-flow-timer seconds;
}
```

Hierarchy Level

```
[edit protocols openflow]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Configure an OpenFlow virtual switch.

Options

switch-name—User-configured identifier for the OpenFlow virtual switch. The identifier can include a maximum of 60 characters.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

controller (Protocols OpenFlow) | 178

default-action (Protocols OpenFlow) | 180

interfaces (Protocols OpenFlow) | 182

openflow (Protocols OpenFlow) | 183

traceoptions (Protocols OpenFlow)

Syntax

```
traceoptions {
    file <filename> <files number> <match regular-expression> <size size> <world-readable | no-world-readable>;
    flag flag;
    no-remote-trace;
}
```

Hierarchy Level

```
[edit protocols openflow]
```

Release Information

Statement introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Define tracing operations for OpenFlow.

Default

If you do not include this statement, no OpenFlow-specific tracing operations are performed.

Options

file *filename*—Name of the file to receive the output of the tracing operation. All files are placed in the **/var/log** directory.

Default: /var/log/ofd

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed and compressed to ***trace-file*.0.gz**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0.gz** is renamed ***trace-file*.1.gz**, and ***trace-file*** is renamed and compressed to ***trace-file*.0.gz**. This renaming scheme continues until the maximum number of trace files is reached. After this, the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size by using the **size** option and also a filename.

Range: 2 through 1000 files

Default: 10 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All OpenFlow events.
- **barrier**—OpenFlow barrier events.
- **configuration**—OpenFlow configuration events.
- **filter**—OpenFlow filter events.
- **flow**—OpenFlow flow events.
- **function**—OpenFlow entry and exit events.
- **group**—(Appears only for Juniper Networks devices running OpenFlow v1.3.1 or later) OpenFlow group events.
- **interface**—OpenFlow interface events.
- **nh**—OpenFlow next-hop events.
- **packet-io**—OpenFlow packet in and packet out events.
- **packets**—OpenFlow packet events.
- **statistics**—OpenFlow statistics request and reply events.
- **switch**—OpenFlow switch events including controller connection messages and keepalives, and packets sent to and received from the controller.

match *regular-expression*—(Optional) Log only those lines that match the regular expression.

no-remote-trace—(Optional) Disable tracing and logging operations that track normal operations, error conditions, and packets that are generated by or have passed through the Juniper Networks device.

no-world-readable—(Optional) Disable unrestricted file access, which restricts file access to the owner. This is the default.

size *size*—(Optional) Maximum size of each trace file in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). If you do not specify a unit, the default is bytes. If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option and a filename.

Syntax: ***size*** to specify bytes, ***size*****k** to specify KB, ***size*****m** to specify MB, or ***size*****g** to specify GB

Range: 10,240 through 1,073,741,824 bytes

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

Required Privilege Level

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

RELATED DOCUMENTATION

Understanding Support for OpenFlow on Devices Running Junos OS | 18

openflow (Protocols OpenFlow) | 183

CHAPTER 6

Operational Commands

OpenFlow Operational Mode Commands | **196**

show openflow capability | **197**

show openflow controller | **205**

show openflow filters | **208**

show openflow flows | **212**

show openflow groups | **218**

show openflow interfaces | **222**

show openflow statistics flows | **227**

show openflow statistics groups | **231**

show openflow statistics interfaces | **234**

show openflow statistics packet | **238**

show openflow statistics queue | **240**

show openflow statistics summary | **243**

show openflow statistics tables | **246**

show openflow summary | **250**

OpenFlow Operational Mode Commands

Table 44 on page 196 summarizes the operational mode commands that you can use to monitor and troubleshoot OpenFlow operations on an OpenFlow-enabled device running Junos OS. Commands are listed in alphabetical order.

Table 44: OpenFlow Operational Mode Commands

Command	Task
show openflow capability	Display support information for OpenFlow features, actions, and match conditions on the device.
show openflow controller	Display OpenFlow controller information and status.
show openflow filters	Display information for filters bound to OpenFlow interfaces.
show openflow flows	Display OpenFlow flow information.
show openflow groups	Display OpenFlow groups information.
show openflow interfaces	Display physical characteristics and status information for interfaces participating in OpenFlow.
show openflow statistics flows	Display statistics for OpenFlow flow entries.
show openflow statistics groups	Display statistics for OpenFlow groups.
show openflow statistics interfaces	Display statistics for interfaces participating in OpenFlow.
show openflow statistics packet	Display statistics for packet-in and packet-out actions.
show openflow statistics queue	Display statistics for OpenFlow queues in hardware.
show openflow statistics summary	Display summary statistics for all OpenFlow flows.
show openflow statistics tables	Display statistics for OpenFlow flow tables.
show openflow summary	Display summary information for OpenFlow flows.
show openflow switch	Display OpenFlow message statistics for OpenFlow virtual switches.

show openflow capability

Syntax

```
show openflow capability
<action | feature | match-condition>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display support information for OpenFlow features, actions, and match conditions on the Juniper Networks device.

Options

none—Display support information for all OpenFlow capabilities.

action—(Optional) Display support information for OpenFlow actions.

feature—(Optional) Display support information for OpenFlow features.

match-condition—(Optional) Display support information for OpenFlow match conditions.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196
OpenFlow Support on Juniper Networks Devices | 17

List of Sample Output

show openflow capability on page 201
show openflow capability (OpenFlow 1.3.1) on page 202
show openflow capability action on page 203
show openflow capability feature on page 203
show openflow capability match-condition on page 203
show openflow capability match-condition (OpenFlow 1.3.1) on page 204

Output Fields

Table 45 on page 198 lists the output fields for the **show openflow capability** command. Output fields are listed in the approximate order in which they appear.

Table 45: show openflow capability Output Fields

Field Name	Field Description
Supported Features—Indicates Whether the Juniper Networks Device Supports the Following OpenFlow Features	
Flow statistics	Indicates whether the device supports OpenFlow flow statistics.
Table statistics	Indicates whether the device supports OpenFlow flow table statistics.
Port statistics	Indicates whether the device supports OpenFlow port statistics.
Group statistics	Indicates whether the device supports OpenFlow group statistics. **NOTE:** This field appears only if the device supports OpenFlow v1.3.1 or later.
802.1d spanning tree	Indicates whether the device supports the 802.1D Spanning Tree Protocol.
Reassemble IP fragments	Indicates whether the device supports reassembling IP fragments.
Queue statistics	Indicates whether the device supports OpenFlow queue statistics.
Match IP addresses in ARP pkts	Indicates whether the device supports matching on IP addresses in ARP packets.
Supported Match Conditions—Indicates Whether the Juniper Networks Device Supports the Following OpenFlow Match Conditions	
Switch input port	Indicates whether the device supports matching against the ingress switch port.
VLAN vid	Indicates whether the device supports matching against the VLAN identifier in the outermost VLAN tag.
Ethernet source address	Indicates whether the device supports matching against the Ethernet source address.
Ethernet destination address	Indicates whether the device supports matching against the Ethernet destination address.
Ethernet frame type	Indicates whether the device supports matching against the Ethernet frame type.

Table 45: show openflow capability Output Fields *(continued)*

Field Name	Field Description
IP protocol	Indicates whether the device supports matching against the IP protocol or lower 8 bits of the ARP opcode.
TCP/UDP source port	Indicates whether the device supports matching against the TCP or UDP source port.
TCP/UDP destination port	Indicates whether the device supports matching against the TCP or UDP destination port.
IP(v4) source address	Indicates whether the device supports matching against the IPv4 source address. **NOTE:** **IPv4** appears only if the device supports OpenFlow v1.3.1 or later.
IP(v4) destination address	Indicates whether the device supports matching against the IPv4 destination address. **NOTE:** **IPv4** appears only if the device supports OpenFlow v1.3.1 or later.
IPv6 source address	Indicates whether the device supports matching against the IPv6 source address. **NOTE:** This field appears only if the device supports OpenFlow v1.3.1 or later.
IPv6 destination address	Indicates whether the device supports matching against the IPv6 destination address. **NOTE:** This field appears only if the device supports OpenFlow v1.3.1 or later.
VLAN priority	Indicates whether the device supports matching against the VLAN priority in the outermost VLAN tag.
IP ToS (DSCP field)	Indicates whether the device supports matching against the IPv4 ToS bits.
Supported Actions—Indicates Whether the Juniper Networks Device Supports the Following OpenFlow Actions	
Output to switch port	Indicates whether the device supports forwarding the packet to a specified port.
Set the 802.1q VLAN id	Indicates whether the device supports the optional Modify-Field action to modify the existing 802.1Q VLAN ID of the outermost VLAN tag in the frame header or to add a new header with the VLAN ID if none exists.

Table 45: show openflow capability Output Fields *(continued)*

Field Name	Field Description
Set the 802.1q priority	Indicates whether the device supports the optional Modify-Field action to modify the existing 802.1Q VLAN priority of the outermost VLAN tag in the frame header or to add a new header with the VLAN priority if none exists.
Strip the 802.1q header	Indicates whether the device supports the optional Modify-Field action to remove the outermost VLAN header in the frame.
Ethernet source address	Indicates whether the device supports the optional Modify-Field action to modify the Ethernet source address field in the frame header.
Ethernet destination address	Indicates whether the device supports the optional Modify-Field action to modify the Ethernet destination address field in the frame header.
IP source address	Indicates whether the device supports the optional Modify-Field action to modify the IP source address field and update the checksum in the packet header.
IP destination address	Indicates whether the device supports the optional Modify-Field action to modify the IP destination address field and update the checksum in the packet header.
IP ToS (DSCP)	Indicates whether the device supports the optional Modify-Field action to modify the IPv4 ToS field in the packet header.
TCP/UDP source port	Indicates whether the device supports the optional Modify-Field action to modify the TCP or UDP source port field and update the checksum in the packet header.
TCP/UDP destination port	Indicates whether the device supports the optional Modify-Field action to modify the TCP or UDP destination port field and update the checksum in the packet header.
Output to queue	Indicates whether the device supports the optional Enqueue action to set the queue ID for the packet.
Execute group	Indicates whether the device supports a group action to be executed. **NOTE:** This field appears only if the device supports OpenFlow v1.3.1 or later.

Sample Output

show openflow capability

```
user@host> show openflow capability
```

```
Openflowd platform feature support:
Flow statistics:    Yes
Table statistics:   Yes
Port statistics:    Yes
802.1d spanning tree:    No
Reassemble IP fragments:    No
Queue statistics:    Yes
Match IP addresses in ARP pkts:    No

Openflowd platform match condition support:
Switch input port:    Yes
VLAN vid:    Yes
Ethernet source address:    Yes
Ethernet destination address:  Yes
Ethernet frame type: Yes
IP protocol:    Yes
TCP/UDP source port:    Yes
TCP/UDP destination port:   Yes
IP source address:    Yes
IP destination address: Yes
VLAN priority:    Yes
IP ToS (DSCP field):    Yes

Openflowd platform action support:
Output to switch port:    Yes
Set the 802.1q VLAN id    Yes
Set the 802.1q priority:    No
Strip the 802.1q header:    Yes
Ethernet source address:    No
Ethernet destination address:    No
IP source address:    No
IP destination address:    No
IP ToS (DSCP):    No
TCP/UDP source port:    No
TCP/UDP destination port:    No
Output to queue:    No
```

show openflow capability (OpenFlow 1.3.1)

`user@host>` **show openflow capability**

```
Openflowd platform feature support:
Flow statistics:    Yes
Table statistics:   Yes
Port statistics:    Yes
Group statistics:    Yes
802.1d spanning tree:    No
Reassemble IP fragments:    No
Queue statistics:    Yes
Match IP addresses in ARP pkts:    No

Openflowd platform match condition support:
Switch input port:    Yes
VLAN vid:    Yes
Ethernet source address:    Yes
Ethernet destination address:  Yes
Ethernet frame type: Yes
IP protocol:    Yes
TCP/UDP source port:    Yes
TCP/UDP destination port:   Yes
IPv4 source address:    Yes
IPv4 destination address: Yes
IPv6 source address:    Yes
IPv6 destination address: Yes
VLAN priority:    Yes
IP ToS (DSCP field):    Yes

Openflowd platform action support:
Output to switch port:    Yes
Set the 802.1q VLAN id    Yes
Set the 802.1q priority:    No
Strip the 802.1q header:    Yes
Ethernet source address:    No
Ethernet destination address:    No
IP source address:    No
IP destination address:    No
IP ToS (DSCP):    No
TCP/UDP source port:    No
TCP/UDP destination port:    No
Output to queue:    No
Execute Group:    Yes
```

show openflow capability action

```
user@host> show openflow capability action
```

```
Openflowd platform action support:
Output to switch port:    Yes
Set the 802.1q VLAN id    Yes
Set the 802.1q priority:    No
Strip the 802.1q header:    Yes
Ethernet source address:    No
Ethernet destination address:    No
IP source address:    No
IP destination address:    No
IP ToS (DSCP):    No
TCP/UDP source port:    No
TCP/UDP destination port:    No
Output to queue:    No
```

show openflow capability feature

```
user@host> show openflow capability feature
```

```
Openflowd platform feature support:
Flow statistics:    Yes
Table statistics:   Yes
Port statistics:    Yes
802.1d spanning tree:    No
Reassemble IP fragments:    No
Queue statistics:    Yes
Match IP addresses in ARP pkts:    No
```

show openflow capability match-condition

```
user@host> show openflow capability match-condition
```

```
Openflowd platform match condition support:
Switch input port:    Yes
VLAN vid:    Yes
Ethernet source address:    Yes
Ethernet destination address:  Yes
Ethernet frame type: Yes
IP protocol:    Yes
TCP/UDP source port:    Yes
TCP/UDP destination port:   Yes
```

```
IP source address:    Yes
IP destination address: Yes
VLAN priority:    Yes
IP ToS (DSCP field):    Yes
```

show openflow capability match-condition (OpenFlow 1.3.1)

`user@host>` **show openflow capability match-condition**

```
Openflowd platform match condition support:
Switch input port:    Yes
VLAN vid:    Yes
Ethernet source address:    Yes
Ethernet destination address:  Yes
Ethernet frame type: Yes
IP protocol:    Yes
TCP/UDP source port:    Yes
TCP/UDP destination port:   Yes
IPv4 source address:    Yes
IPv4 destination address: Yes
IPv6 source address:    Yes
IPv6 destination address: Yes
VLAN priority:    Yes
IP ToS (DSCP field):    Yes
```

show openflow controller

Syntax

```
show openflow controller
<address address>
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display OpenFlow controller information and connection status. OpenFlow controllers are configured at the **[edit protocols openflow switch *switch-name*]** hierarchy level.

Options

none—Display information about all configured controllers.

address *address*—(Optional) Display information about the controller at the specified IP address.

switch *switch-name*—(Optional) Display information about controllers associated with the specified virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | 196
- controller (Protocols OpenFlow) | 178
- OpenFlow Support on Juniper Networks Devices | 17
- Understanding the OpenFlow Version Negotiation Between the Controller and Devices Running Junos OS | 28

List of Sample Output

show openflow controller (OpenFlow 1.3.1) on page 206
show openflow controller address (OpenFlow 1.3.1) on page 207
show openflow controller switch (OpenFlow 1.3.1) on page 207

Output Fields

Table 46 on page 206 lists the output fields for the **show openflow controller** command. Output fields are listed in the approximate order in which they appear.

Table 46: show openflow controller Output Fields

Field Name	Field Description
Controller socket	Socket on the controller to which the OpenFlow virtual switch connects.
Controller IP address	IP address of the OpenFlow controller.
Controller protocol	Protocol used by the switch to initiate a connection with the controller.
Controller port	Port on the controller to which the OpenFlow virtual switch connects.
Controller connection state	Status of the connection between the OpenFlow virtual switch and the controller.
Number of connection attempt	Number of connection attempts made by the virtual switch to the controller.
Controller role	User-configured role of the controller.
Negotiated version	A numerical value that represents the OpenFlow version that is negotiated between the Junos OS device and the OpenFlow controller during the initial connection. **NOTE:** This field appears only if the Junos OS device supports OpenFlow version 1.3.1 or later. .
Negotiated version	A numerical value that represents the OpenFlow version that is negotiated between the Junos OS device and the OpenFlow controller during the initial connection. **NOTE:** This field appears only if the Junos OS device supports OpenFlow version 1.3.1 or later.

Sample Output

show openflow controller (OpenFlow 1.3.1)

```
user@host> show openflow controller
```

```
Openflowd controller information:
Controller socket: 15
Controller IP address: 198.51.100.174
```

```
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 5
Controller role: equal
Negotiated version: 4
```

show openflow controller address (OpenFlow 1.3.1)

`user@host>` **show openflow controller address 198.51.100.174**

```
Openflowd controller information:
Controller socket: 15
Controller IP address: 198.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 5
Controller role: equal
Negotiated version: 4
```

show openflow controller switch (OpenFlow 1.3.1)

`user@host>` **show openflow controller switch OFswitch1**

```
Openflowd controller information:
Controller socket: 15
Controller IP address: 198.51.100.174
Controller protocol: tcp
Controller port: 6633
Controller connection state: up
Number of connection attempt: 5
Controller role: equal
Negotiated version: 4
```

show openflow filters

Syntax

```
show openflow filters
<interface interface-name>
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display information for filters bound to OpenFlow interfaces.

Options

none—Display information for all filters that are bound to OpenFlow interfaces.

interface *interface-name*—(Optional) Display information for the filter bound to the specified OpenFlow interface. The interface name must include the logical unit number.

switch *switch-name*—(Optional) Display information for filters bound to the interfaces configured under the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196
Understanding OpenFlow Flows and Filters on Devices Running Junos OS | 29

List of Sample Output

show openflow filters on page 209
show openflow filters interface on page 210
show openflow filters switch on page 210

Output Fields

Table 47 on page 209 lists the output fields for the **show openflow filters** command. Output fields are listed in the approximate order in which they appear.

Table 47: show openflow filters Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch associated with the interface to which the filter is bound.
Number of filters	Number of filters bound to OpenFlow interfaces on the virtual switch.
Default action	Default action executed for packets that do not match any existing flow entries. Values are **PACKET IN** or **DROP**.
Filter name	Filter identifier consisting of the concatenation of the interface name (including the logical unit number) and an internally assigned switch ID.
Filter index	Auto-generated string that identifies the filter.
Number of terms	Number of terms in the filter. Each term consists of match conditions and actions.
Number of priorities	Number of unique active flow priorities in the filter.
Term name	Filter term identifier, which consists of the filter name (interface name and switch ID), the flow priority, and a sequence number.
Priority ID	Flow entry priority. Higher priority terms are installed above lower priority terms.
Flow ID	Flow identifier associated with that flow entry.
Number of packets	Number of packets that have matched a filter term. A filter term is equivalent to a flow entry.
Number of bytes	Number of bytes that have matched a filter term. A filter term is equivalent to a flow entry.

Sample Output

show openflow filters

```
user@host> show openflow filters
```

```
Switch                  Filter      Number of  Number of  Number of
Name                    Index       terms      priorities packets
OFswitch1               96468992    0          0          0
                        96468993    0          0          0
                        96468994    0          0          0
                        96468995    0          0          0
                        96468996    1          1          7928017621
```

show openflow filters interface

user@host> **show openflow filters interface ge-1/1/7.0**

```
Switch Name: OFswitch1
Filter name: ge-1/1/7.0_0
Filter index: 96468996
Number of terms: 1            Number of priorities: 1

Term name: ge-1/1/7.0_0:32766^OF:1
Priority ID: 32766                      Flow ID:        16842752
Number of packets:7941332819            Number of bytes:476479969140
```

show openflow filters switch

user@host> **show openflow filters switch OFswitch1**

```
Switch Name: OFswitch1
Number of filters: 5            Default action: PACKET IN

Filter name: ge-1/1/0.0_0
Filter index: 96468992
Number of terms: 0            Number of priorities: 0

Filter name: ge-1/1/1.0_0
Filter index: 96468993
Number of terms: 0            Number of priorities: 0

Filter name: ge-1/1/2.0_0
Filter index: 96468994
Number of terms: 0            Number of priorities: 0

Filter name: ge-1/1/3.0_0
Filter index: 96468995
```

```
Number of terms: 0            Number of priorities: 0

Filter name: ge-1/1/7.0_0
Filter index: 96468996
Number of terms: 1            Number of priorities: 1
Priority  Flow                   Number of              Number of
ID        ID                     packets                bytes
32768     16842752               7941332819             476479969140
```

show openflow flows

Syntax

```
show openflow flows
<brief | detail | summary>
<flow-id>
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display information about traffic flows that match filters on OpenFlow interfaces.

Options

none—(Same as **brief**) Display information for all flows.

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

flow-id—(Optional) Display information only for the specified flow.

switch *switch-name*—(Optional) Display information only for the flows on the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196
OpenFlow Support on Juniper Networks Devices | 17

List of Sample Output

show openflow flows switch brief on page 214
show openflow flows switch detail on page 214
show openflow flows switch detail (OpenFlow 1.3.1, IPv6 Flow) on page 215
show openflow flows switch summary on page 215
show openflow flows brief (Specific Flow) on page 215
show openflow flows detail (Specific Flow) on page 216
show openflow flows detail (Specific Flow, OpenFlow 1.3.1, IPv6 Flow) on page 216

show openflow flows detail (Specific Flow, OpenFlow 1.3.1, Destination MAC Address Rewrite Action) on page 217
show openflow flows summary (Specific Flow) on page 217

Output Fields

Table 48 on page 213 lists the output fields for the **show openflow flows** command. Output fields are listed in the approximate order in which they appear.

Table 48: show openflow flows Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch on which the flow resides.
Number of flows	Number of active flow entries associated with that OpenFlow virtual switch.
Flow name	Flow descriptor.
Table ID	Identifier for the flow table from which the flow originated.
Flow ID	Flow identifier associated with that flow entry.
Number of packets	Number of packets that have matched the flow entry.
Priority	Flow entry priority. Packets match higher priority entries before matching lower priority entries.
Idle timeout	Number of seconds after which the flow entry is removed from the flow table provided there are no matching packets.
Hard timeout	Number of seconds after which the flow entry is removed from the flow table regardless of the number of matching packets.
Cookie	An identifier specified by the OpenFlow controller when the flow is installed in the flow table. Cookies are used to filter flows for flow modification and delete operations. **NOTE:** This field appears only if the device running Junos OS supports OpenFlow version 1.3.1 or later.

Table 48: show openflow flows Output Fields *(continued)*

Field Name	Field Description
Match	Configured match conditions against which the incoming packet is compared. **NOTE:** In the OpenFlow 1.3.1 sample output, the match fields that are populated for an IPv6 flow apply to MX Series routers only.
Action	Set of actions (for OpenFlow v1.0) or flow instructions (for OpenFlow v1.3.1) applied to a packet when it matches the flow entry.
Number of match	Number of match conditions against which the incoming packet is compared.
Number of action	Number of actions (for OpenFlow v1.0) or flow instructions (for OpenFlow v1.3.1) that are applied to a packet when it matches the flow entry.

Sample Output

show openflow flows switch brief

`user@host>` **show openflow flows switch OFswitch1 brief**

```
Switch                Flow    Number of packets    Priority Number of Number of
Name                  ID                                    match    action
OFswitch1             16842752 8075372509          32768    1         1
```

show openflow flows switch detail

`user@host>` **show openflow flows switch OFswitch1 detail**

```
Flow name: flow-16842752
Table ID: 1     Flow ID: 16842752
Priority: 32768   Idle timeout(in sec):0        Hard timeout(in sec): 0
Cookie: 0
Match: Input port: 45549
       Ethernet src addr: wildcard
       Ethernet dst addr: wildcard
       Input vlan id: wildcard          Input VLAN priority: wildcard
       Ether type: wildcard
```

```
       IP ToS: wildcard                IP protocol: wildcard
       IP src addr: wildcard           IP dst addr: wildcard
       Source port: wildcard           Destination port: wildcard
Action: Output port 41350,
```

show openflow flows switch detail (OpenFlow 1.3.1, IPv6 Flow)

`user@host>` **show openflow flows switch OFswitch1 detail**

```
Flow name: flow-67174400
Table ID: 1     Flow ID: 67174400
Priority: 8       Idle timeout(in sec):1800     Hard timeout(in sec): 1800
Cookie: 3333
Match: Input port: 2
       Ethernet src addr: aa:bb:cc:11:22:33
       Ethernet dst addr: aa:bb:cc:11:22:34
       Input vlan id: 50               Input VLAN priority: 3
       Ether type: 0x86dd
       IP ToS: 0x2                     IP protocol: 0x6
       IPv4 src addr: NA
       IPv4 dst addr: NA
       IPv6 src addr: 2001::/64
       IPv6 dst addr: 1001::1/128
       Source port: 8                  Destination port: 9
Action: Set_Field[VLAN_ID] 601, Output port 4, Set_Field[VLAN_ID] 701, Output port
 6,
```

show openflow flows switch summary

`user@host>` **show openflow flows switch OFswitch1 summary**

```
Switch Name                                                    Number of Flows
OFswitch1                                                      1
```

show openflow flows brief (Specific Flow)

`user@host>` **show openflow flows 16842752 brief**

```
Switch               Flow    Number of packets    Priority Number of Number of
```

```
Name                  ID                                    match    action
OFswitch1             16842752 8056139439          32768    1         1
```

show openflow flows detail (Specific Flow)

`user@host>` **show openflow flows 16842752 detail**

```
Flow name: flow-16842752
Table ID: 1     Flow ID: 16842752
Priority: 32768   Idle timeout(in sec):0        Hard timeout(in sec): 0
Cookie: 0
Match: Input port: 45549
       Ethernet src addr: wildcard
       Ethernet dst addr: wildcard
       Input vlan id: wildcard          Input VLAN priority: wildcard
       Ether type: wildcard
       IP ToS: wildcard                 IP protocol: wildcard
       IP src addr: wildcard            IP dst addr: wildcard
       Source port: wildcard            Destination port: wildcard
Action: Output port 41350,
```

show openflow flows detail (Specific Flow, OpenFlow 1.3.1, IPv6 Flow)

`user@host>` **show openflow flows 67174400 detail**

```
Flow name: flow-67174400
Table ID: 1     Flow ID: 67174400
Priority: 8       Idle timeout(in sec):1800     Hard timeout(in sec): 1800
Cookie: 3333
Match: Input port: 2
       Ethernet src addr: aa:bb:cc:11:22:33
       Ethernet dst addr: aa:bb:cc:11:22:34
       Input vlan id: 50                Input VLAN priority: 3
       Ether type: 0x86dd
       IP ToS: 0x2                      IP protocol: 0x6
       IPv4 src addr: NA
       IPv4 dst addr: NA
       IPv6 src addr: 2001::/64
       IPv6 dst addr: 1001::1/128
       Source port: 8                   Destination port: 9
Action: Set_Field[VLAN_ID] 601, Output port 4, Set_Field[VLAN_ID] 701, Output port
 6,
```

show openflow flows detail (Specific Flow, OpenFlow 1.3.1, Destination MAC Address Rewrite Action)

`user@host>` **show openflow flows 16842752 detail**

```
Flow name: flow-16842752
Table ID: 1     Flow ID: 16842752
Priority: 1       Idle timeout(in sec):0        Hard timeout(in sec): 0
Cookie: 3333
Match: Input port: 1
       Ethernet src addr: wildcard
       Ethernet dst addr: wildcard
       Input vlan id: wildcard          Input VLAN priority: wildcard
       Ether type: wildcard
       IP ToS: wildcard                 IP protocol: wildcard
       IPv4 src addr: wildcard
       IPv4 dst addr: wildcard
       IPv6 src addr: wildcard
       IPv6 dst addr: wildcard
       Source port: wildcard            Destination port: wildcard
Action: Dst MAC ab:cd:ef:12:34:56, Output port 2,
```

show openflow flows summary (Specific Flow)

`user@host>` **show openflow flows 16842752 summary**

```
Flow name: flow-16842752
Number of packets: 8066495711
```

show openflow groups

Syntax

```
show openflow groups
<brief | details | summary>
<group-id>
```

Release Information

Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.
Command introduced in Junos OS Release 14.2.
Command introduced in Junos OS Release 14.2 for EX Series switches.

Description

Display information about OpenFlow groups. Groups are supported only on Juniper Networks devices running OpenFlow v1.3.1 or later.

Options

none—Display information for all groups.

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

group-id—(Optional) Display information about the specified group only.

Required Privilege Level

admin

RELATED DOCUMENTATION

- Understanding How the OpenFlow Group Action Works | 33
- OpenFlow Operational Mode Commands | 196
- show openflow statistics groups | 231

List of Sample Output

show openflow groups on page 219
show openflow groups brief on page 220
show openflow groups detail on page 220
show openflow groups summary on page 220
show openflow groups (Specific Group) on page 220
show openflow groups brief (Specific Group) on page 221
show openflow groups detail (Specific Group) on page 221
show openflow groups summary (Specific Group) on page 221

Output Fields

Table 49 on page 219 describes the output fields for the **show openflow groups** command. Table 49 on page 219 lists the output fields in the approximate order in which they are displayed in the sample output.

Table 49: show openflow groups Output Fields

Field Name	Field Description
Group ID	Unique identifier assigned to a group by the OpenFlow controller.
Type	Group type, which can be either All or Indirect.
Number of Buckets	Number of buckets for a particular group. A group can have 0 to 32 buckets.
Number of Flow Reference	Number of flow entries that point to a particular group.
Bucket	Information about each bucket for a particular group.
Actions	Set of action(s) applied to a packet when it matches the flow entry.
Switch Name	User-configured identifier for the OpenFlow virtual switch on which the flow resides.
Number of Groups	Number of groups that currently exist in the OpenFlow virtual switch.
Flow	Identifier associated with a particular flow entry.

Sample Output

show openflow groups

```
user@host> show openflow groups
```

```
Group                Type          Number of          Number of Flow
ID                                 Buckets            Reference
50                   All           2                  1
51                   All           2                  1
60                   Indirect      1                  0
```

show openflow groups brief

```
user@host> show openflow groups brief
```

```
Group                Type          Number of           Number of Flow
ID                                 Buckets             Reference
50                   All           2                   1
51                   All           2                   1
60                   Indirect      1                   0
```

show openflow groups detail

```
user@host> show openflow groups detail
```

```
Group Id: 50                    Type: All
Bucket Bucket 1
      Actions: VLAN ID 2022, Output port 2,
Bucket Bucket 2
      Actions: VLAN ID 3022, Output port 4,

Group Id: 51                    Type: All
Bucket Bucket 3
      Actions: VLAN ID 2001, Output port 1,
Bucket Bucket 4
      Actions: VLAN ID 3001, Output port 3,

Group Id: 60                    Type: Indirect
Bucket Bucket 5
      Actions: VLAN ID 2060, Output port 3,
```

show openflow groups summary

```
user@host> show openflow groups summary
```

```
Switch Name                                                    Number of Groups
OF-ex92k                                                        3
```

show openflow groups (Specific Group)

```
user@host> show openflow groups 50
```

```
Group                Type          Number of           Number of Flow
ID                                 Buckets             Reference
50                   All           2                   1
```

show openflow groups brief (Specific Group)

`user@host>` **show openflow groups 50 brief**

```
Group                Type          Number of           Number of Flow
ID                                 Buckets             Reference
50                   All           2                   1
```

show openflow groups detail (Specific Group)

`user@host>` **show openflow groups 50 detail**

```
Group Id: 50                    Type: All
Bucket 1
      Actions: VLAN ID 2022, Output port 2,
Bucket 2
      Actions: VLAN ID 3022, Output port 4,
Flow 570710622208
```

show openflow groups summary (Specific Group)

`user@host>` **show openflow groups 50 summary**

```
Group                Type          Number of           Number of Flow
ID                                 Buckets             Reference
50                   All           2                   1
```

show openflow interfaces

Syntax

```
show openflow interfaces
<interface-name>
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display physical characteristics and status information for interfaces participating in OpenFlow.

Options

none—Display information for all interfaces participating in OpenFlow.

interface-name—(Optional) Display information only for the specified interface. Specify the interface name including the logical unit number—for example, ge-1/1/0.0.

switch *switch-name*—(Optional) Display information only for those interfaces configured under the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196

show openflow filters | 208

show openflow flows | 212

show openflow statistics interfaces | 234

List of Sample Output

show openflow interfaces on page 223
show openflow interfaces (Specific Interface) on page 224
show openflow interfaces switch on page 225

Output Fields

Table 50 on page 223 lists the output fields for the **show openflow interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 50: show openflow interfaces Output Fields

Field Name	Field Description
Switch name	User-configured identifier for the OpenFlow virtual switch to which the interface is bound.
Interface Name	Name of the logical interface.
Interface port number	Port identifier associated with the OpenFlow interface.
Interface Hardware Address	Media access control (MAC) address of the interface.
Interface speed	Speed and duplex mode of the interface.
Interface Auto-Negotiation	Autonegotiation status: **Enabled** or **Disabled**.
Interface media type	Media type of the interface. For example, copper or fiber.
Interface state	Current state of the interface.

Sample Output

show openflow interfaces

`user@host>` **show openflow interfaces**

```
Switch name: OFswitch1
Interface Name: ge-1/1/2.0
Interface port number: 41507
Interface Hardware Address: 00:00:5e:00:53:b4
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/3.0
Interface port number: 44383
```

```
Interface Hardware Address: 00:00:5e:00:53:b5
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/1.0
Interface port number: 41350
Interface Hardware Address: 00:00:5e:00:53:b7
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/7.0
Interface port number: 45549
Interface Hardware Address: 00:00:5e:00:53:b6
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/0.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:b2
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up
```

show openflow interfaces (Specific Interface)

`user@host>` **show openflow interfaces ge-1/1/0.0**

```
Switch name: OFswitch1
Interface Name: ge-1/1/0.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:b2
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
```

```
Interface media type: Fiber
Interface state: Up
```

show openflow interfaces switch

user@host> **show openflow interfaces switch OFswitch1**

```
Switch name: OFswitch1
Interface Name: ge-1/1/2.0
Interface port number: 41507
Interface Hardware Address: 00:00:5e:00:53:b4
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/3.0
Interface port number: 44383
Interface Hardware Address: 00:00:5e:00:53:b5
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/1.0
Interface port number: 41350
Interface Hardware Address: 00:00:5e:00:53:b7
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
Interface Name: ge-1/1/7.0
Interface port number: 45549
Interface Hardware Address: 00:00:5e:00:53:b6
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up

Switch name: OFswitch1
```

```
Interface Name: ge-1/1/0.0
Interface port number: 44538
Interface Hardware Address: 00:00:5e:00:53:b2
Interface speed: 1Gb Full-duplex
Interface Auto-Negotiation: Enabled
Interface media type: Fiber
Interface state: Up
```

show openflow statistics flows

Syntax

```
show openflow statistics flows
<flow-id>
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display statistics for OpenFlow flows.

Options

none—Display flow statistics for all flows for all OpenFlow virtual switches.

flow-id—(Optional) Display flow statistics only for the specified flow.

switch *switch-name*—(Optional) Display flow statistics only for the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196

show openflow flows | 212

show openflow statistics interfaces | 234

show openflow statistics packet | 238

show openflow statistics tables | 246

List of Sample Output

show openflow statistics flows on page 228
show openflow statistics flows (OpenFlow 1.3.1) on page 229
show openflow statistics flows (Specific Flow) on page 229
show openflow statistics flows (Specific Flow, OpenFlow 1.3.1) on page 229
show openflow statistics flows switch on page 230
show openflow statistics flows switch (OpenFlow 1.3.1) on page 230

Output Fields

Table 51 on page 228 lists the output fields for the **show openflow statistics flows** command. Output fields are listed in the approximate order in which they appear.

Table 51: show openflow statistics flows Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch on which the flow resides.
Table ID	Identifier for the flow table from which the flow originated.
Flow ID	OpenFlow flow entry identifier.
Duration(in sec)	Number of seconds the flow has been active.
Duration(in nsec)	Number of nanoseconds the flow has been active beyond the value of **Duration(in sec)**.
Priority	Flow entry priority. Packets match higher priority entries before matching lower priority entries.
Idle timeout	Number of seconds after which the flow entry is removed from the flow table provided there are no matching packets.
Hard timeout	Number of seconds after which the flow entry is removed from the flow table regardless of the number of matching packets.
Number of packets	Number of packets that have matched the flow entry.
Number of bytes	Number of bytes that have matched the flow entry.
Match	Fields against which the incoming packet is compared.
Action	Set of actions applied to a packet when it matches the flow entry.

Sample Output

show openflow statistics flows

```
user@host> show openflow statistics flows
```

```
Switch Name: OFswitch1
Table ID: 1     Flow ID: 16842752
Duration(in sec): 58772          Duration(in nsec): 215702000
Priority: 32768   Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 8745275026
Number of bytes:   524716501560
Match: IN_PORT,
Action: OUTPUT,
```

show openflow statistics flows (OpenFlow 1.3.1)

`user@host>` **show openflow statistics flows**

```
Switch Name: 100
Table ID: 1     Flow ID: 67174400
Duration(in sec): 51539          Duration(in nsec): 3961483296
Priority: 1000    Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 179965839
Number of bytes:   14757198798
Match: IN_PORT, DL_VLAN, DL_TYPE, IPv6_SRC, IPv6_DST,
Action: SET_FIELD[VLAN_ID], OUTPUT,
```

show openflow statistics flows (Specific Flow)

`user@host>` **show openflow statistics flows 16842752**

```
Switch Name: OFswitch1
Table ID: 1     Flow ID: 16842752
Duration(in sec): 58803          Duration(in nsec): 4127548296
Priority: 32768   Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 8749713419
Number of bytes:   524982805140
Match: IN_PORT,
Action: OUTPUT,
```

show openflow statistics flows (Specific Flow, OpenFlow 1.3.1)

`user@host>` **show openflow statistics flows 67174400**

```
Switch Name: 100
Table ID: 1     Flow ID: 67174400
Duration(in sec): 51539          Duration(in nsec): 3961483296
```

```
Priority: 1000    Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 179965839
Number of bytes:   14757198798
Match: IN_PORT, DL_VLAN, DL_TYPE, IPv6_SRC, IPv6_DST,
Action: SET_FIELD[VLAN_ID], OUTPUT,
```

show openflow statistics flows switch

`user@host>` **show openflow statistics flows switch OFswitch1**

```
Switch Name: OFswitch1
Table ID: 1     Flow ID: 16842752
Duration(in sec): 58829          Duration(in nsec): 4124448296
Priority: 32768   Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 8752672358
Number of bytes:   525160341480
Match: IN_PORT,
Action: OUTPUT,
```

show openflow statistics flows switch (OpenFlow 1.3.1)

`user@host>` **show openflow statistics flows switch 100**

```
Switch Name: 100
Table ID: 1     Flow ID: 67174400
Duration(in sec): 51539          Duration(in nsec): 3961483296
Priority: 1000    Idle timeout(in sec):0        Hard timeout(in sec): 0
Number of packets: 179965839
Number of bytes:   14757198798
Match: IN_PORT, DL_VLAN, DL_TYPE, IPv6_SRC, IPv6_DST,
Action: SET_FIELD[VLAN_ID], OUTPUT,
```

show openflow statistics groups

Syntax

```
show openflow statistics groups
<group-id>
```

Release Information

Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.
Command introduced in Junos OS Release 14.2.
Command introduced in Junos OS Release 14.2 for EX Series switches.

Description

Display statistics for OpenFlow groups. Groups are supported only on Juniper Networks devices running OpenFlow v1.3.1 or later.

Options

none—Display statistics for all groups defined on all OpenFlow virtual switches.

group-id—(Optional) Display statistics only for the specified group.

Required Privilege Level

admin

RELATED DOCUMENTATION

- Understanding How the OpenFlow Group Action Works | **33**
- OpenFlow Operational Mode Commands | **196**
- show openflow groups | **218**

List of Sample Output

show openflow statistics group on page 232
show openflow statistics group (Specific Group) on page 233

Output Fields

Table 52 on page 232 lists the output fields for the **show openflow statistics groups** command. Output fields are listed in the approximate order in which they appear.

Table 52: show openflow statistics groups Output Fields

Field	Description
Switch Name	User-configured identifier for the OpenFlow virtual switch in which the groups resides.
Group ID	Unique identifier assigned to the group by the OpenFlow controller.
Ref Count	Number of flow entries that reference the group.
Number of packets (group)	Number of packets handled by the group.
Number of bytes (group)	Number of bytes handled by the group.
Duration(in sec)	Number of seconds the group has been active.
Duration(in nsec)	Number of nanoseconds the group has been active beyond the **Duration(in sec)**
Bucket *number*	Statistics for a particular bucket.
Number of packets (bucket)	Number of packets handled by a bucket in the group.
Number of bytes (bucket)	Number of bytes handled by a bucket in the group.

For a group with the group type of all, the values specified in the **Number of packets** (group) and **Number of bytes** (group) fields are usually the same as those specified in the **Number of packets** (bucket) and **Number of bytes** (bucket) fields because all buckets in the group are executed.

Sample Output

show openflow statistics group

```
user@host> show openflow statistics groups
```

```
Switch Name: OF-ex92k
Group ID: 50                    Ref Count: 1
Number of packets: 62161
Number of bytes:   7956608
Duration(in sec): 22687          Duration(in nsec): 4255381296
    Bucket 0
```

```
      Number of packets: 62161
      Number of bytes:   7956608
    Bucket 1
      Number of packets: 62161
      Number of bytes:   7956608

Switch Name: OF-ex92k
Group ID: 51                    Ref Count: 1
Number of packets: 0
Number of bytes:   0
Duration(in sec): 22673          Duration(in nsec): 8549000
    Bucket 0
      Number of packets: 0
      Number of bytes:   0
    Bucket 1
      Number of packets: 0
      Number of bytes:   0
...
```

show openflow statistics group (Specific Group)

`user@host>` **show openflow statistics groups 50**

```
Switch Name: OF-ex92k
Group ID: 50                    Ref Count: 1
Number of packets: 64886
Number of bytes:   8305408
Duration(in sec): 22789          Duration(in nsec): 586200000
    Bucket 0
      Number of packets: 64886
      Number of bytes:   8305408
    Bucket 1
      Number of packets: 64886
      Number of bytes:   8305408
```

show openflow statistics interfaces

Syntax

```
show openflow statistics interfaces
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display statistics for interfaces participating in OpenFlow.

Options

none—Display statistics for all interfaces participating in OpenFlow for all configured OpenFlow virtual switches.

switch *switch-name*—(Optional) Display statistics only for those interfaces on the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | 196
- show openflow interfaces | 222
- show openflow statistics flows | 227
- show openflow statistics tables | 246

List of Sample Output

show openflow statistics interfaces on page 236

Output Fields

Table 53 on page 235 lists the output fields for the **show openflow statistics interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 53: show openflow statistics interfaces Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch to which the interface is bound.
Interface Name	Name of the logical interface.
Port Number	Port identifier associated with the OpenFlow interface.
Num of rx pkts	Number of packets received on the OpenFlow interface.
Num of tx pkts	Number of packets transmitted on the OpenFlow interface.
Num of rx bytes	Number of bytes received on the OpenFlow interface.
Num of tx bytes	Number of bytes transmitted on the OpenFlow interface.
Num of rx error	Number of receive errors.
Num of tx error	Number of transmit errors.
Number of packets dropped by RX	Number of packets dropped by the ingress interface.
Number of packets dropped by TX	Number of packets dropped by the egress interface.
Number of rx frame error	Number of packets with frame alignment errors.
Number of rx overrun error	Number of packets with RX overrun.
Number of CRC error	Number of CRC errors.
Number of collisions	Number of Ethernet collisions.

Sample Output

show openflow statistics interfaces

`user@host>` **show openflow statistics interfaces**

```
Switch Name: OFswitch1
Interface Name: ge-1/1/2.0       Port Number: 41507
Num of rx pkts: 0                          Num of tx pkts: 1372301
Num of rx bytes: 0                         Num of tx bytes: 88665532
Num of rx error: 0                         Num of tx error:0
Number of packets dropped by RX: 0
Number of packets dropped by TX: 0
Number of rx frame error:        0
Number of rx overrun error:      0
Number of CRC error:             0
Number of collisions:            0

Switch Name: OFswitch1
Interface Name: ge-1/1/3.0       Port Number: 44383
Num of rx pkts: 0                          Num of tx pkts: 1372285
Num of rx bytes: 0                         Num of tx bytes: 88664476
Num of rx error: 0                         Num of tx error:0
Number of packets dropped by RX: 0
Number of packets dropped by TX: 0
Number of rx frame error:        0
Number of rx overrun error:      0
Number of CRC error:             0
Number of collisions:            0

Switch Name: OFswitch1
Interface Name: ge-1/1/1.0       Port Number: 41350
Num of rx pkts: 0                          Num of tx pkts: 8776241344
Num of rx bytes: 0                         Num of tx bytes: 526580807026
Num of rx error: 0                         Num of tx error:0
Number of packets dropped by RX: 0
Number of packets dropped by TX: 0
Number of rx frame error:        0
Number of rx overrun error:      0
Number of CRC error:             0
Number of collisions:            0

Switch Name: OFswitch1
Interface Name: ge-1/1/7.0       Port Number: 45549
Num of rx pkts: 8840952127                 Num of tx pkts: 1047701
```

```
Num of rx bytes: 530457127620              Num of tx bytes: 69187816
Num of rx error: 0                         Num of tx error:0
Number of packets dropped by RX: 0
Number of packets dropped by TX: 0
Number of rx frame error:        0
Number of rx overrun error:      0
Number of CRC error:             0
Number of collisions:            0

Switch Name: OFswitch1
Interface Name: ge-1/1/0.0       Port Number: 44538
Num of rx pkts: 0                          Num of tx pkts: 1372031
Num of rx bytes: 0                         Num of tx bytes: 88647712
Num of rx error: 0                         Num of tx error:0
Number of packets dropped by RX: 0
Number of packets dropped by TX: 0
Number of rx frame error:        0
Number of rx overrun error:      0
Number of CRC error:             0
Number of collisions:            0
```

show openflow statistics packet

Syntax

```
show openflow statistics packet (in | out)
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display statistics for packet-in and packet-out (send-packet) actions.

Options

none—Display statistics for all OpenFlow virtual switches.

switch *switch-name*—(Optional) Display statistics only for the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | 196

List of Sample Output

show openflow statistics packet in on page 239
show openflow statistics packet out on page 239
show openflow statistics packet out switch on page 239

Output Fields

Table 54 on page 238 lists the output fields for the **show openflow statistics packet** command. Output fields are listed in the approximate order in which they appear.

Table 54: show openflow statistics packet Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.

Table 54: show openflow statistics packet Output Fields *(continued)*

Field Name	Field Description
Rx packets	Number of packets received by the OpenFlow virtual switch that have been sent to the OpenFlow controller. The switch includes the packet in the data portion of an OFPT_PACKET_IN message.
Tx packets	Number of packets sent by the OpenFlow controller to an egress interface. The controller includes the packet in the data portion of an OFPT_PACKET_OUT message.
Drop packets	Number of dropped packets.

Sample Output

show openflow statistics packet in

```
user@host> show openflow statistics packet in
```

```
Openflow packet-in statistics information:
Switch Name                                              Rx packets  Drop packets
OFswitch1                                                1044137     0
```

show openflow statistics packet out

```
user@host> show openflow statistics packet out
```

```
Openflow packet-out statistics information:
Switch Name                                              Tx packets  Drop packets
OFswitch1                                                5260759     0
```

show openflow statistics packet out switch

```
user@host> show openflow statistics packet out switch OFswitch1
```

```
Openflow packet-out statistics information:
Switch Name                                              Tx packets  Drop packets
OFswitch1                                                5260759     0
```

show openflow statistics queue

Syntax

```
show openflow statistics queue
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display statistics for hardware queues for interfaces participating in OpenFlow.

Options

none—Display queue statistics for all interfaces participating in OpenFlow.

interface *interface-name*—(Optional) Display queue statistics only for the specified interface. Specify the interface name including the logical unit number—for example, ge-1/1/0.0

Required Privilege Level

admin

RELATED DOCUMENTATION

OpenFlow Operational Mode Commands | **196**

show openflow statistics flows | **227**

show openflow statistics tables | **246**

List of Sample Output

show openflow statistics queue on page 241
show openflow statistics queue interface on page 242

Output Fields

Table 55 on page 241 lists the output fields for the **show openflow statistics queue** command. Output fields are listed in the approximate order in which they appear.

Table 55: show openflow statistics queue Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.
Port No	Port identifier associated with the OpenFlow interface.
Queue Id	Priority queue identifier.
TX bytes	Number of bytes transmitted through the queue.
TX packets	Number of packets transmitted through the queue.
Tx errors	Number of packets dropped by the queue due to overrun.

Sample Output

show openflow statistics queue

`user@host>` **show openflow statistics queue**

```
Openflow queue statistics information:
Switch Name            Port No Queue Id   TX bytes     TX packets   Tx errors
OFswitch1              41507  0           115327076    1372459      0
OFswitch1              41507  1           0            0            0
OFswitch1              41507  2           0            0            0
OFswitch1              41507  3           0            0            0
OFswitch1              41507  4           0            0            0
OFswitch1              41507  5           0            0            0
OFswitch1              41507  6           0            0            0
OFswitch1              41507  7           0            0            0
OFswitch1              44383  0           115325732    1372443      0
OFswitch1              44383  1           0            0            0
OFswitch1              44383  2           0            0            0
OFswitch1              44383  3           0            0            0
OFswitch1              44383  4           0            0            0
OFswitch1              44383  5           0            0            0
OFswitch1              44383  6           0            0            0
OFswitch1              44383  7           0            0            0
OFswitch1              41350  0           752072717540 8953246155   0
OFswitch1              41350  1           0            0            0
OFswitch1              41350  2           0            0            0
```

```
OFswitch1              41350  3           0            0            0
OFswitch1              41350  4           0            0            0
OFswitch1              41350  5           0            0            0
OFswitch1              41350  6           0            0            0
OFswitch1              41350  7           0            0            0
OFswitch1              45549  0           88060496     1047859      0
OFswitch1              45549  1           0            0            0
OFswitch1              45549  2           0            0            0
OFswitch1              45549  3           0            0            0
OFswitch1              45549  4           0            0            0
OFswitch1              45549  5           0            0            0
OFswitch1              45549  6           0            0            0
OFswitch1              45549  7           0            0            0
OFswitch1              44538  0           115304396    1372189      0
OFswitch1              44538  1           0            0            0
OFswitch1              44538  2           0            0            0
OFswitch1              44538  3           0            0            0
OFswitch1              44538  4           0            0            0
OFswitch1              44538  5           0            0            0
OFswitch1              44538  6           0            0            0
OFswitch1              44538  7           0            0            0
```

show openflow statistics queue interface

`user@host>` **show openflow statistics queue interface ge-1/1/2.0**

```
Openflow queue statistics information:
Switch Name            Port No Queue Id   TX bytes     TX packets   Tx errors
OFswitch1              41507  0           115327076    1372459      0
OFswitch1              41507  1           0            0            0
OFswitch1              41507  2           0            0            0
OFswitch1              41507  3           0            0            0
OFswitch1              41507  4           0            0            0
OFswitch1              41507  5           0            0            0
OFswitch1              41507  6           0            0            0
OFswitch1              41507  7           0            0            0
```

show openflow statistics summary

Syntax

```
show openflow statistics summary
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display summary statistics for all installed OpenFlow flow entries for all OpenFlow virtual switches.

Options

This command has no options.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | 196
- show openflow statistics flows | 227
- show openflow statistics tables | 246
- show openflow summary | 250

List of Sample Output

show openflow statistics summary on page 244

Output Fields

Table 56 on page 243 lists the output fields for the **show openflow statistics summary** command. Output fields are listed in the approximate order in which they appear.

Table 56: show openflow statistics summary Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.
Port Number	Port identifier associated with the OpenFlow interface.

Table 56: show openflow statistics summary Output Fields *(continued)*

Field Name	Field Description
Number of RX packets	Number of packets received on the OpenFlow interface.
Number of TX packets	Number of packets transmitted on the OpenFlow interface.
Num of packets dropped by RX	Number of packets dropped by the ingress interface.
Flow ID	Flow identifier associated with that flow entry.
Number of packets	Number of packets that have matched the flow entry.
Duration (in sec)	Number of seconds the flow has been active.
Priority	Flow entry priority. Packets match higher priority entries before matching lower priority entries.
Idle Timeout	Number of seconds after which the flow entry is removed from the flow table provided there are no matching packets.
Hard Timeout	Number of seconds after which the flow entry is removed from the flow table regardless of the number of matching packets.

Sample Output

show openflow statistics summary

```
user@host> show openflow statistics summary
```

```
Switch    Port   Number of RX packets    Number of TX packets    Num of packets
Name      Number                                                 dropped by RX
OFswitch1    41507 0                     1372609                 0
OFswitch1    44383 0                     1372593                 0
OFswitch1    41350 0                     9119477900              0
OFswitch1    45549 9184188377            1048009                 0
OFswitch1    44538 0                     1372339                 0
```

```
Switch          Flow   Number of packets     Duration  Priority Idle    Hard
Name            ID                           (in sec)           Timeout Timeout
OFswitch1       16842752 9117212928          61278       32768  0       0
```

show openflow statistics tables

Syntax

```
show openflow statistics tables
<switch switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display statistics for OpenFlow flow tables.

Options

none—Display statistics for flow tables on all OpenFlow virtual switches.

switch *switch-name*—(Optional) Display statistics only for flow tables on the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | **196**
- show openflow statistics flows | **227**
- show openflow statistics interfaces | **234**
- show openflow statistics summary | **243**

List of Sample Output

show openflow statistics tables on page 247
show openflow statistics tables (OpenFlow 1.3.1) on page 248
show openflow statistics tables switch on page 248
show openflow statistics tables switch (OpenFlow 1.3.1) on page 248

Output Fields

Table 57 on page 247 lists the output fields for the **show openflow statistics tables** command. Output fields are listed in the approximate order in which they appear.

Table 57: show openflow statistics tables Output Fields

Field Name	Field Description
Table Name	String identifier for the OpenFlow flow table.
Table id	Numeric identifier for the OpenFlow flow table.
Supported wildcards	Wildcards supported by the flow table.
Max number of entries	Maximum number of entries supported in the flow table.
Number of active entries	Number of active entries in the flow table.
Number of idle timeout entries	Number of entries in the flow table that have been removed because the idle timeout expired and no packets matched those entries.
Number of hard timeout entries	Number of entries in the flow table that have been removed because the hard timeout expired.
Number of flow delete entries	Number of entries in the flow table that have been removed in response to controller requests.
Number of flow add entries	Number of entries in the flow table that have been added in response to controller requests.
Number of flow modify entries	Number of entries in the flow table that have been modified in response to controller requests.
Number of total delete entries	Number of entries in the flow table that have been removed for any reason.

Sample Output

show openflow statistics tables

`user@host>` **show openflow statistics tables**

```
Table name: Default flow table                  Table id:1
Supported wildcards: IN_PORT, DL_VLAN, DL_SRC, DL_DST, DL_TYPE, NW_PROTO, TP_SRC,
```

```
 TP_DST, NW_SRC, NW_DST, DL_VLAN_PCP, NW_TOS,
Max number of entries: 65535              Number of active entries: 1
Number of idle timeout entries: 0
Number of hard timeout entries: 0
Number of flow delete entries:  0
Number of flow add entries:     1
Number of flow modify entries:  0
Number of total delete entries: 0
```

show openflow statistics tables (OpenFlow 1.3.1)

```
user@host> show openflow statistics tables
```

```
Table name: Default flow table                  Table id:1
Supported wildcards: IN_PORT, DL_VLAN, DL_SRC, DL_DST, DL_TYPE, NW_PROTO, TP_SRC,
 TP_DST, IPv4_SRC, IPv4_DST, IPv6_SRC, IPv6_DST, DL_VLAN_PCP, NW_TOS,
Number of active entries: 7
Number of entries used: 0
Number of idle timeout entries: 0
Number of hard timeout entries: 0
Number of flow delete entries:  0
Number of flow add entries:     7
Number of flow modify entries:  0
Number of total delete entries: 0
```

show openflow statistics tables switch

```
user@host> show openflow statistics tables switch OFswitch1
```

```
Table name: Default flow table                  Table id:1
Supported wildcards: IN_PORT, DL_VLAN, DL_SRC, DL_DST, DL_TYPE, NW_PROTO, TP_SRC,
 TP_DST, NW_SRC, NW_DST, DL_VLAN_PCP, NW_TOS,
Max number of entries: 65535              Number of active entries: 1
Number of idle timeout entries: 0
Number of hard timeout entries: 0
Number of flow delete entries:  0
Number of flow add entries:     1
Number of flow modify entries:  0
Number of total delete entries: 0
```

show openflow statistics tables switch (OpenFlow 1.3.1)

```
user@host> show openflow statistics tables switch 100
```

```
Table name: Default flow table                  Table id:1
Supported wildcards: IN_PORT, DL_VLAN, DL_SRC, DL_DST, DL_TYPE, NW_PROTO, TP_SRC,
 TP_DST, IPv4_SRC, IPv4_DST, IPv6_SRC, IPv6_DST, DL_VLAN_PCP, NW_TOS,
Number of active entries: 7
Number of entries used: 0
Number of idle timeout entries: 0
Number of hard timeout entries: 0
Number of flow delete entries:  0
Number of flow add entries:     7
Number of flow modify entries:  0
Number of total delete entries: 0
```

show openflow summary

Syntax

```
show openflow summary
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display summary information for OpenFlow including the number of configured virtual switches, controllers, interfaces, and flows.

Options

This command has no options.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | 196
- show openflow statistics summary | 243
- show openflow switch | 252

Output Fields

Table 58 on page 250 lists the output fields for the **show openflow summary** command. Output fields are listed in the approximate order in which they appear.

Table 58: show openflow summary Output Fields

Field Name	Field Description
Number of switches	Total number of configured OpenFlow virtual switches.
Number of controllers	Total number of configured OpenFlow controllers.

Table 58: show openflow summary Output Fields *(continued)*

Field Name	Field Description
Number of interfaces	Number of logical interfaces participating in OpenFlow.
Number of active flow entries	Number of active entries in the flow table.

Sample Output

show openflow summary

`user@host>` **show openflow summary**

```
Number of switches:            1
Number of controllers:         1
Number of interfaces:          5
Number of active flow entries: 1
```

show openflow switch

Syntax

```
show openflow switch
<switch-name>
```

Release Information

Command introduced in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.

Description

Display OpenFlow message statistics for OpenFlow virtual switches.

Options

none—Display information for all OpenFlow virtual switches.

switch *switch-name*—(Optional) Display information only for the specified OpenFlow virtual switch.

Required Privilege Level

admin

RELATED DOCUMENTATION

- OpenFlow Operational Mode Commands | 196
- show openflow statistics tables | 246
- show openflow summary | 250

List of Sample Output

show openflow switch on page 254
show openflow switch (Specific OpenFlow Virtual Switch) on page 254

Output Fields

Table 59 on page 252 lists the output fields for the **show openflow switch** command. Output fields are listed in the approximate order in which they appear.

Table 59: show openflow switch Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.

Table 59: show openflow switch Output Fields *(continued)*

Field Name	Field Description
Switch ID	Device identifier for the OpenFlow virtual switch.
Switch DPID	Data path ID uniquely identifying the OpenFlow instance. This value is a concatenation of the switch ID for the virtual switch and the management port MAC address.
Flow mod received	Number of Modify Flow Entry messages (OFPT_FLOW_MOD) received from the controller.
Vendor received	Number of messages with vendor-specific extensions.
Packets sent	Number of packets sent to the controller.
Packets received	Number of packets received from the controller.
Echo req sent	Number of Echo Request messages (OFPT_ECHO_REQUEST) sent to the controller.
Echo req received	Number of Echo Request messages (OFPT_ECHO_REQUEST) received from the controller.
Echo reply sent	Number of Echo Reply messages (OFPT_ECHO_REPLY) sent to the controller.
Echo reply received	Number of Echo Reply messages (OFPT_ECHO_REPLY) received from the controller.
Port Status sent	Number of Port Status messages (OFPT_PORT_STATUS) sent to the controller.
Port mod received	Number of Port Modification messages (OFPT_PORT_MOD) received from the controller.
Barrier request	Number of Barrier Request messages (OFPT_BARRIER_REQUEST) received from the controller.
Barrier reply	Number of Barrier Reply messages (OFPT_BARRIER_REPLY) sent to the controller.
Error msg sent	Number of error messages (OFPT_ERROR) sent to the controller.

Table 59: show openflow switch Output Fields *(continued)*

Field Name	Field Description
Error msg received	Number of error messages (OFPT_ERROR) received from the controller.

Sample Output

show openflow switch

```
user@host> show openflow switch
```

```
Switch Name:        OFswitch1
Switch ID:          0                   Switch DPID:    00:00:00:00:5e:00:53:d0
Flow mod received:  4                   Vendor received:      0
Packets sent:       1048258             Packets received:     1089664
Echo req sent:      4115                Echo req received:    0
Echo reply sent:    0                   Echo reply received:  4115
Port Status sent:   1                   Port mod received:    0
Barrier request:    0                   Barrier reply:        0
Error msg sent:     1                   Error msg received:   0
```

show openflow switch (Specific OpenFlow Virtual Switch)

```
user@host> show openflow switch OFswitch1
```

```
Switch Name:        OFswitch1
Switch ID:          0                   Switch DPID:    00:00:00:00:5e:00:53:d0
Flow mod received:  4                   Vendor received:      0
Packets sent:       1048259             Packets received:     1089675
Echo req sent:      4116                Echo req received:    0
Echo reply sent:    0                   Echo reply received:  4116
Port Status sent:   1                   Port mod received:    0
Barrier request:    0                   Barrier reply:        0
Error msg sent:     1                   Error msg received:   0
```