Junos® OS for EX9200 Switches

OpenFlow for EX9200 Switches

Release
14.1

Published: 2014-09-29

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

Junos® OS for EX9200 Switches OpenFlow for EX9200 Switches
Release 14.1

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

List of Tables

About the Documentation

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 http://www.juniper.net/techpubs/.

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 http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- EX Series

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 the *CLI User Guide*.

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**

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

Convention	Description	Examples
`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*
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		
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**.

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

Convention	Description	Examples
> (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, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC 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 http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

OpenFlow Overview

OpenFlow Support on Devices Running Junos OS

Table 3 on page 3 lists the devices running Junos OS that support OpenFlow. For each device, the table lists the supported OpenFlow versions, the initial Junos OS Release that must be installed for OpenFlow support, and the corresponding OpenFlow software package for that product and release. The OpenFlow software package release must match the Junos OS release of the device on which the software is installed.

Table 3: OpenFlow Support on Devices Running Junos OS

Product	OpenFlow version	Junos OS Release	Software Package
EX4550 Ethernet Switches	v1.0	13.2X51-D20	jsdn-powerpc
EX9200 Ethernet Switches	v1.0	13.3R1	jsdn-i386
MX80 3D Universal Edge Router	v1.0	13.3R1	jsdn-powerpc
MX240, MX480, MX960 3D Universal Edge Routers	v1.0	13.3R1	jsdn-i386

Table 4 on page 4 lists support for various OpenFlow features on devices running Junos OS that support OpenFlow.

Table 4: OpenFlow Features supported on Devices Running Junos OS

Product	Hybrid Interfaces	Multi-VLAN Support	OpenFlow over MPLS
EX4550 Ethernet Switches	No	No	No
EX9200 Ethernet Switches	Yes	Yes	No
MX80 3D Universal Edge Router	Yes	Yes	Yes
MX240, MX480, MX960 3D Universal Edge Routers	Yes	Yes	Yes

Related Documentation

- Understanding Support for OpenFlow on Devices Running Junos OS on page 5
- Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS on page 7
- Installing Support for OpenFlow on Devices Running Junos OS on page 27
- OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS on page 15

Supported Open Standards

Junos OS substantially supports the following open standards.

- OpenFlow specification, *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 15.

  Junos OS differs from the specification in the following ways:

  - Junos OS only supports 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 as per 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 deletes normal flow entries, and after 30 seconds, purges flow entries that are installed in hardware.

The features discussed in the indicated sections of the specification are not supported:

- Encryption through TLS connection (section 4.4)
- 802.1D Spanning Tree Protocol (section 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 section 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)
- Switch configuration using OFPT_SET_CONFIG (section 5.3.2)
- 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)

Related Documentation

- OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS on page 15
- Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS on page 7

Understanding Support for OpenFlow on Devices Running Junos OS

OpenFlow Overview

OpenFlow is an open standard that enables you to control traffic and run experimental protocols in an existing network 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 that are applied to matching packets.

You can configure certain devices running the 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.

For more information about OpenFlow, see the Open Networking Foundation website at https://www.opennetworking.org/ .

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 enter or exit from OpenFlow-enabled interfaces, and MAC address learning is disabled on these interfaces.

Interfaces participating in OpenFlow must be configured as Layer 2 interfaces. To configure the 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 only be configured 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 devices running Junos OS, you can only configure a single logical unit 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 could 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 Devices Running Junos OS on page 3
- Installing Support for OpenFlow on Devices Running Junos OS on page 27
- Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS on page 9
- Understanding OpenFlow Flows and Filters on Devices Running Junos OS on page 11
- Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS on page 7
- OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS on page 15

Understanding OpenFlow Operation and Forwarding Actions on Devices Running Junos OS

This topic summarizes the OpenFlow features and forwarding actions supported on devices running Junos OS. 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 15.

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 only enter or exit from 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_UNSUPPORTED 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 5 on page 7 summarizes the general features supported on devices running Junos OS that support OpenFlow. For information about support on specific platforms, see "OpenFlow Support on Devices Running Junos OS" on page 3.

Table 5: OpenFlow Support on Devices Running Junos OS

Feature	Support
OpenFlow v1.0	Supported.

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

Feature	Support
OpenFlow virtual switch	1 OpenFlow virtual switch.
Controller	1 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 deletes normal flow entries, and after 30 seconds, purges flow entries that are installed in hardware.
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
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 only configure Ethernet interfaces 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.

OpenFlow Forwarding Actions

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.

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 only flood packets out OpenFlow interfaces configured under that specific virtual switch excluding the ingress OpenFlow interface.

Related Documentation

- Understanding Support for OpenFlow on Devices Running Junos OS on page 5
- OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS on page 15
- Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS on page 9
- Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS on page 12

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

On devices running the 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 or fxp0 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 highest OpenFlow protocol version supported by the sender. If the first packet received by the switch is not an OFPT_HELLO message, the switch tears down 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 session establishment, the controller sends the switch a feature request message to request 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 tears down 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 on page 5

Understanding OpenFlow Flows and Filters on Devices Running Junos OS

OpenFlow flows comprise a set of match conditions against which packets are compared, the set of actions that are applied to matching packets, and a flow priority. The match conditions combined with the flow priority uniquely identify a flow. 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 auto-generated 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. 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 are mapped to the filter term **then** statement. Depending on the flow action, the **then** statement might include actions for forwarding the packet to the next hop or OpenFlow controller, or discarding the packet.

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 only modify or delete 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.

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 on 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 12. 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 are considered invalid. The invalid flows are removed from the Packet Forwarding Engine but are indefinitely retained by the Routing Engine until such time as 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 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 on page 5
- Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS on page 12

Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS

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 and others that are set through the Junos OS configuration. OpenFlow flow entry timers include the idle

timeout, the hard timeout, and the purge flow timer. Table 6 on page 13 summarizes the various OpenFlow flow timers.

Table 6: OpenFlow Flow Entry Timers

Timer	Configured Through	Range (Seconds)
Idle timeout	Controller	0, 11 through 65535
Hard timeout	Controller	0 through 65535
Purge flow timer	Junos OS **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 the flow is removed from the flow table and the hardware provided there are no matching packets. The hard timeout is the number of seconds after which the flow is removed from the flow table and the hardware regardless of the number of matching packets.

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 is removed at that point only if there are no matching packets. Otherwise, the flow 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 65535 seconds. If the controller sets the idle timeout to 0, the flow entry does not idle time out. The hard timeout value can range from 0 through 65535 seconds. If the controller sets the hard timeout to 0, the flow entry does not 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 Junos OS 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 flows 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 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 flows.

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 flows from the hardware but indefinitely retains the flow entries in the flow table. If the logical interface later becomes available, the flows 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 flows 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 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 on page 5
- purge-flow-timer on page 56

Understanding OpenFlow Barrier Messages on Devices Running Junos OS

OpenFlow-enabled devices running 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 on page 5
- Understanding OpenFlow Flows and Filters on Devices Running Junos OS on page 11

OpenFlow v1.0 Compliance Matrix for Devices Running Junos OS

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

- Table 7 on page 15 lists support for message types.
- Table 8 on page 17 lists support for port structure flags.
- Table 9 on page 18 lists support for match conditions.
- Table 10 on page 19 lists support for wildcards.
- Table 11 on page 19 lists support for flow actions.
- Table 12 on page 20 lists support for flow actions in Send Packet messages (OFPT_PACKET_OUT).
- Table 13 on page 21 lists support for statistics.
- Table 14 on page 21 lists support for features.

Table 7 on page 15 lists the support for OpenFlow v1.0 message types.

Table 7: Junos OS Support for OpenFlow v1.0 Message Types

Section	Specification	MX Series	EX9200
5.1	OFPT_HELLO	Supported	Supported
	OFPT_ERROR	Supported	Supported
	OFPT_ECHO_REQUEST	Supported	Supported
	OFPT_ECHO_REPLY	Supported	Supported
	OFPT_VENDOR	Not supported	Not supported
	OFPT_FEATURES_REQUEST	Supported	Supported

Table 7: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	MX Series	EX9200
	OFPT_FEATURES_REPLY:	Supported	Supported
	Datapath ID	Supported	Supported
	N_buffers	-1	-1
	N_tables	1	1
	OFPC_FLOW_STATS	Supported	Supported
	OFPC_TABLE_STATS	Supported	Supported
	OFPC_PORT_STATS	Supported	Supported
	OFPC_STP	Not supported	Not supported
	OFPC_IP_REASM	Not supported	Not supported
	OFPC_QUEUE_STATS	Supported	Supported
	OFPC_ARP_MATCH_IP	Not supported	Not supported
	OFPT_GET_CONFIG_REQUEST	Supported	Supported
	OFPT_GET_CONFIG_REPLY	Supported	Supported
	OFPT_SET_CONFIG	Not supported	Not supported
	OFPT_PACKET_IN	Supported	Supported
	OFPT_PACKET_IN with buffer_id	Not supported	Not supported
	OFPT_FLOW_REMOVED	Supported	Supported
	OFPT_PORT_STATUS	Supported	Supported
	OFPT_PACKET_OUT	Supported	Supported
	OFPT_PACKET_OUT with buffer_id	Not supported	Not supported
	OFPT_FLOW_MOD:	Supported	Supported
	OFPFC_ADD	Supported	Supported
	OFPFC_ADD with OFPFF_CHECK_OVERLAP	Supported	Supported
	OFPFC_MODIFY	Supported	Supported
	OFPFC_MODIFY_STRICT	Supported	Supported
	OFPFC_DELETE	Supported	Supported
	OFPFC_DELETE_STRICT	Supported	Supported
	OFPT_FLOW_MOD with buffer_id	Not supported	Not supported
	OFPT_PORT_MOD	Not supported	Not supported
	OFPT_STATS_REQUEST	Supported	Supported
	OFPT_STATS_REPLY See Table 13 on page 21	Supported	Supported
	OFPT_BARRIER_REQUEST	Supported	Supported
	OFPT_BARRIER_REPLY	Supported	Supported

Table 7: Junos OS Support for OpenFlow v1.0 Message Types *(continued)*

Section	Specification	MX Series	EX9200
	OFPT_QUEUE_GET_CONFIG_REQUEST	Not supported	Not supported
	OFPT_QUEUE_GET_CONFIG_REPLY	Not supported	Not supported

Table 8 on page 17 lists the support for OpenFlow v1.0 port structure flags.

Table 8: 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
	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

Table 8: Junos OS Support for OpenFlow v1.0 Port Structure Flags *(continued)*

Section	Specification	MX Series	EX9200
	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 9 on page 18 lists the support for OpenFlow v1.0 match conditions.

Table 9: 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

Table 9: Junos OS Support for OpenFlow v1.0 Match Conditions *(continued)*

Section	Specification	MX Series	EX9200
	tp_dst (TCP/UDP destination port)	Supported	Supported
	Match all 12 tuples or a combination of tuples	Supported	Supported

Table 10 on page 19 lists the support for OpenFlow v1.0 wildcards.

Table 10: 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
	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 wild cards set. Match entire 12 tuple.	Supported	Supported

Table 11 on page 19 lists the support for OpenFlow v1.0 flow actions.

Table 11: 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

Table 11: Junos OS Support for OpenFlow v1.0 Flow Actions *(continued)*

Section	Specification	MX Series	EX9200
	OFPAT_STRIP_VLAN	Supported	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
	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 12 on page 20 lists the support for OpenFlow v1.0 flow actions in Send Packet messages (OFPT_PACKET_OUT).

Table 12: 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 12: 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 13 on page 21 lists the support for OpenFlow v1.0 statistics.

Table 13: 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 14 on page 21 lists the support for OpenFlow v1.0 features.

Table 14: 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
	Flow Priority	Supported	Supported

Related Documentation

- Understanding Support for OpenFlow on Devices Running Junos OS on page 5

CHAPTER 2

OpenFlow Hybrid Interfaces

Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS

On Juniper Networks EX9200 Ethernet Switches and on MX Series 3D Universal Edge Routers 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 egress 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 on page 5
- *Configuring OpenFlow Hybrid Interfaces on MX Series Routers*
- Configuring OpenFlow Hybrid Interfaces on EX9200 Switches on page 37
- *Binding VLAN IDs to Logical Interfaces*

PART 2

Configuration

CHAPTER 3

OpenFlow Basic Configuration

Installing Support for OpenFlow on Devices Running Junos OS

You can add support for OpenFlow on a device running Junos OS by copying the software package to your device and then installing the package. The software package is identified by the jsdn prefix, and the filename string begins with the following format:

jsdn-*packageID-release*

where:

- *packageID* identifies the devices running Junos OS on which you can install the package.
- *release* identifies the release, for example, 13.3. The jsdn software release and the Junos OS release of the device on which it is installed must match.

For information about OpenFlow support on devices running Junos OS and the corresponding installation package for that device, see "OpenFlow Support on Devices Running Junos OS" on page 3.

To install the jsdn software package on a device running Junos OS:

1. Download the software package to the device.
2. If you previously installed the jsdn software package, remove the existing package using the **request system software delete** operational mode command.

   user@host> request system software delete *existing-jsdn-package*
3. Install the new software package using the **request system software add** operational mode command.

   user@host> request system software add *path-to-jsdn-package*

Related Documentation

- OpenFlow Support on Devices Running Junos OS on page 3
- Understanding Support for OpenFlow on Devices Running Junos OS on page 5

Configuring Support for OpenFlow on EX9200 Switches

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 37.

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

To configure the OpenFlow interfaces:

1. Specify the desired VLAN tagging 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, 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 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 on page 5
- Installing Support for OpenFlow on Devices Running Junos OS on page 27
- OpenFlow Operational Mode Commands on page 63
- **openflow (Protocols OpenFlow) on page 54**
- [edit protocols openflow] Hierarchy Level on page 49

Example: Enabling OpenFlow on EX9200 Switches

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 fxp0 interface and is reachable from the OpenFlow controller IP address.

Overview

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 OpenFlow traffic only will either enter or exit 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

Confirm that the configuration is working properly.

- Verifying the OpenFlow Controller Connection on page 34
- Verifying the OpenFlow Interfaces on page 34

Verifying the OpenFlow Controller Connection

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: 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 on page 5
- Installing Support for OpenFlow on Devices Running Junos OS on page 27
- Configuring Support for OpenFlow on EX9200 Switches on page 28
- OpenFlow Operational Mode Commands on page 63
- **openflow (Protocols OpenFlow) on page 54**
- [edit protocols openflow] Hierarchy Level on page 49

CHAPTER 4

OpenFlow Hybrid Interfaces

Configuring OpenFlow Hybrid Interfaces on EX9200 Switches

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 on page 39
- Understanding OpenFlow Hybrid Interfaces on Devices Running Junos OS on page 23
- *OpenFlow Feature Guide*

Example: Configuring OpenFlow Hybrid Interfaces on EX9200 Switches

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

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 15 on page 41 summarizes the logical interfaces, traffic types, and associated VLAN IDs.

Table 15: Summary of Logical Interfaces

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
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

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

Confirm that the configuration is working properly.

- Verifying the OpenFlow Controller Connection on page 46
- Verifying the OpenFlow Interfaces on page 46

Verifying the OpenFlow Controller Connection

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: 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 on page 23
- Configuring OpenFlow Hybrid Interfaces on EX9200 Switches on page 37
- *OpenFlow Feature Guide*

CHAPTER 5

Configuration Statements

[edit protocols openflow] Hierarchy Level

```
protocols {
  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;
          }
        }
      }
```

Related Documentation

- Understanding Support for OpenFlow on Devices Running Junos OS on page 5
- openflow (Protocols OpenFlow) on page 54
- OpenFlow Operational Mode Commands on page 63

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.

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 on page 5
- controller (Protocols OpenFlow) on page 51
- protocol (Protocols OpenFlow) on page 55
- switch (Protocols OpenFlow) on page 58

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.

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.

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 on page 5
- Understanding the Virtual Switch Connection to the OpenFlow Controller on Devices Running Junos OS on page 9
- address (Protocols OpenFlow) on page 50
- protocol (Protocols OpenFlow) on page 55
- role (Protocols OpenFlow) on page 57
- switch (Protocols OpenFlow) on page 58

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.

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 on page 5
- Understanding OpenFlow Flows and Filters on Devices Running Junos OS on page 11
- openflow (Protocols OpenFlow) on page 54
- switch (Protocols OpenFlow) on page 58

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.

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 on page 5
- controller (Protocols OpenFlow) on page 51

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.

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 32640

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 on page 5
- openflow (Protocols OpenFlow) on page 54
- switch (Protocols OpenFlow) on page 58

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.

Description Configure support for OpenFlow on a device running Junos OS. To configure OpenFlow, the device must be running a 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.

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 Devices Running Junos OS on page 3
- Understanding Support for OpenFlow on Devices Running Junos OS on page 5
- OpenFlow Operational Mode Commands on page 63

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.

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 65535

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 on page 5
- address (Protocols OpenFlow) on page 50
- controller (Protocols OpenFlow) on page 51
- protocol (Protocols OpenFlow) on page 55

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.

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.

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 on page 5
- controller (Protocols OpenFlow) on page 51
- port (Protocols OpenFlow) on page 55

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.

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 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 on page 5
- Understanding OpenFlow Flow Entry Timers on Devices Running Junos OS on page 12

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.

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 on page 5
- controller (Protocols OpenFlow) on page 51

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.

Description Configure an OpenFlow virtual switch.

Options ***switch-name***—User-configured identifier for the OpenFlow virtual switch. The identifier must be 60 characters or less.

The remaining statements are explained separately.

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 on page 5
- controller (Protocols OpenFlow) on page 51
- default-action (Protocols OpenFlow) on page 52
- interfaces (Protocols OpenFlow) on page 53
- openflow (Protocols OpenFlow) on page 54

traceoptions (Protocols OpenFlow)

Syntax

```
traceoptions {
  file <filename> <files number> <match regular-expression> <size size>
    <world-readable | no-world-readable>;
  flag flag;
}
```

Hierarchy Level [edit protocols openflow]

Release Information Statement introduced in Junos OS Release 13.3.

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. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and 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.
- **configuration**—OpenFlow configuration events.
- **filter**—OpenFlow filter events.
- **flow**—OpenFlow flow events.
- **function**—OpenFlow entry and exit events.
- **interface**—OpenFlow interface 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) Only log lines that match the regular expression.

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: 10240 through 1073741824 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 on page 5
- openflow (Protocols OpenFlow) on page 54

PART 3

Administration

CHAPTER 6

Operational Commands

OpenFlow Operational Mode Commands

Table 16 on page 63 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 16: 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.

Table 16: OpenFlow Operational Mode Commands *(continued)*

Command	Task
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 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.

Description Display support information for OpenFlow features, actions, and match conditions on the 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 on page 63

List of Sample Output
show openflow capability on page 67
show openflow capability action on page 68
show openflow capability feature on page 68
show openflow capability match-condition on page 68

Output Fields Table 17 on page 65 lists the output fields for the **show openflow capability** command. Output fields are listed in the approximate order in which they appear.

Table 17: show openflow capability Output Fields

Field Name	Field Description
Supported Features—Indicates Support for the Following OpenFlow Features	
Flow statistics	Indicates if the switch supports OpenFlow flow statistics.
Table statistics	Indicates if the switch supports OpenFlow flow table statistics.
Port statistics	Indicates if the switch supports OpenFlow port statistics.
802.1d spanning tree	Indicates if the switch supports the 802.1D Spanning Tree Protocol.
Reassemble IP fragments	Indicates if the switch supports reassembling IP fragments.
Queue statistics	Indicates if the switch supports OpenFlow queue statistics.
Match IP addresses in ARP pkts	Indicates if the switch supports matching on IP addresses in ARP packets.

Table 17: show openflow capability Output Fields *(continued)*

Field Name	Field Description
Supported Match Conditions—Indicates Support for the Following OpenFlow Match Conditions	
Switch input port	Displays support for matching against the ingress switch port.
VLAN vid	Displays support for matching against the VLAN identifier in the outermost VLAN tag.
Ethernet source address	Displays support for matching against the Ethernet source address.
Ethernet destination address	Displays support for matching against the Ethernet destination address.
Ethernet frame type	Displays support for matching against the Ethernet frame type.
IP protocol	Displays support for matching against the IP protocol or lower 8 bits of the ARP opcode.
TCP/UDP source port	Displays support for matching against the TCP or UDP source port.
TCP/UDP destination port	Displays support for matching against the TCP or UDP destination port.
IP source address	Displays support for matching against the IP source address.
IP destination address	Displays support for matching against the IP destination address.
VLAN priority	Displays support for matching against the VLAN priority in the outermost VLAN tag.
IP ToS (DSCP field)	Displays support for matching against the IPv4 ToS bits.
Supported Actions—Indicates Support for the Following OpenFlow Actions	
Output to switch port	Displays support for forwarding the packet to a specified port.
Set the 802.1q VLAN id	Displays support for 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.
Set the 802.1q priority	Displays support for 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	Displays support for the optional Modify-Field action to remove the outermost VLAN header in the frame.
Ethernet source address	Displays support for the optional Modify-Field action to modify the Ethernet source address field in the frame header.

Table 17: show openflow capability Output Fields *(continued)*

Field Name	Field Description
Ethernet destination address	Displays support for the optional Modify-Field action to modify the Ethernet destination address field in the frame header.
IP source address	Displays support for the optional Modify-Field action to modify the IP source address field and update the checksum in the packet header.
IP destination address	Displays support for the optional Modify-Field action to modify the IP destination address field and update the checksum in the packet header.
IP ToS (DSCP)	Displays support for the optional Modify-Field action to modify the IPv4 ToS field in the packet header.
TCP/UDP source port	Displays support for 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	Displays support for 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	Displays support for the optional Enqueue action to set the queue ID for the packet.

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 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 controller

Syntax

show openflow controller
<address *address*>
<switch *switch-name*>

Release Information

Command introduced in Junos OS Release 13.3.

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 on page 63
- controller (Protocols OpenFlow) on page 51

List of Sample Output

show openflow controller on page 70
show openflow controller address on page 70
show openflow controller switch on page 70

Output Fields

Table 18 on page 69 lists the output fields for the **show openflow controller** command. Output fields are listed in the approximate order in which they appear.

Table 18: 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.

Table 18: show openflow controller Output Fields *(continued)*

Field Name	Field Description
Number of connection attempt	Number of connection attempts made by the virtual switch to the controller.
Controller role	User-configured role of the controller.

Sample Output

show openflow controller

```
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
```

show openflow controller address

```
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
```

show openflow controller switch

```
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
```

show openflow filters

Syntax

show openflow filters
<**interface** *interface-name*>
<**switch** *switch-name*>

Release Information

Command introduced in Junos OS Release 13.3.

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 on page 63
- Understanding OpenFlow Flows and Filters on Devices Running Junos OS on page 11

List of Sample Output

show openflow filters on page 72
show openflow filters interface on page 72
show openflow filters switch on page 72

Output Fields

Table 19 on page 71 lists the output fields for the **show openflow filters** command. Output fields are listed in the approximate order in which they appear.

Table 19: 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.

Table 19: show openflow filters Output Fields *(continued)*

Field Name	Field Description
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.

Description

Display information for OpenFlow flows.

Options

none—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 on page 63

List of Sample Output

show openflow flows switch brief on page 75
show openflow flows switch detail on page 75
show openflow flows switch summary on page 75
show openflow flows 16842752 brief on page 75
show openflow flows 16842752 detail on page 76
show openflow flows 16842752 summary on page 76

Output Fields

Table 20 on page 74 lists the output fields for the **show openflow flows** command. Output fields are listed in the approximate order in which they appear.

Table 20: 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.

Table 20: show openflow flows Output Fields *(continued)*

Field Name	Field Description
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.
Match	Configured match conditions against which the incoming packet is compared.
Action	Set of actions 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 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
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 summary

```
user@host> show openflow flows switch OFswitch1 summary

Switch Name                                                    Number of Flows
OFswitch1                                                      1
```

show openflow flows 16842752 brief

```
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 16842752 detail

```
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
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 16842752 summary

```
user@host> show openflow flows 16842752 summary
Flow name: flow-16842752
Number of packets: 8066495711
```

show openflow interfaces

Syntax

show openflow interfaces
<*interface-name*>
<**switch** *switch-name*>

Release Information

Command introduced in Junos OS Release 13.3.

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 on page 63
- show openflow filters on page 71
- show openflow flows on page 74
- show openflow statistics interfaces on page 84

List of Sample Output

show openflow interfaces on page 78
show openflow interfaces ge-1/1/0.0 on page 79
show openflow interfaces switch on page 79

Output Fields

Table 21 on page 77 lists the output fields for the **show openflow interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 21: 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.

Table 21: show openflow interfaces Output Fields *(continued)*

Field Name	Field Description
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 ge-1/1/0.0

```
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.

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 on page 63
- show openflow flows on page 74
- show openflow statistics interfaces on page 84
- show openflow statistics packet on page 87
- show openflow statistics tables on page 93

List of Sample Output

show openflow statistics flows on page 82
show openflow statistics flows 16842752 on page 82
show openflow statistics flows switch on page 82

Output Fields

Table 22 on page 81 lists the output fields for the **show openflow statistics flows** command. Output fields are listed in the approximate order in which they appear.

Table 22: 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 **Duration(in sec)**.

Table 22: show openflow statistics flows Output Fields *(continued)*

Field Name	Field Description
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 16842752

```
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 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 interfaces

Syntax **show openflow statistics interfaces**
<switch *switch-name*>

Release Information Command introduced in Junos OS Release 13.3.

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 on page 63
- show openflow interfaces on page 77
- show openflow statistics flows on page 81
- show openflow statistics tables on page 93

List of Sample Output show openflow statistics interfaces on page 85

Output Fields Table 23 on page 84 lists the output fields for the **show openflow statistics interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 23: 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.

Table 23: show openflow statistics interfaces Output Fields *(continued)*

Field Name	Field Description
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.

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 on page 63

List of Sample Output
show openflow statistics packet in on page 87
show openflow statistics packet out on page 88
show openflow statistics packet out switch on page 88

Output Fields Table 24 on page 87 lists the output fields for the **show openflow statistics packet** command. Output fields are listed in the approximate order in which they appear.

Table 24: show openflow statistics packet Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.
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.

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 on page 63
- show openflow statistics flows on page 81
- show openflow statistics tables on page 93

List of Sample Output
show openflow statistics queue on page 89
show openflow statistics queue interface on page 90

Output Fields Table 25 on page 89 lists the output fields for the **show openflow statistics queue** command. Output fields are listed in the approximate order in which they appear.

Table 25: 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.

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 on page 63
- show openflow statistics flows on page 81
- show openflow statistics tables on page 93
- show openflow summary on page 95

List of Sample Output show openflow statistics summary on page 92

Output Fields Table 26 on page 91 lists the output fields for the **show openflow statistics summary** command. Output fields are listed in the approximate order in which they appear.

Table 26: 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.
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.

Table 26: show openflow statistics summary Output Fields *(continued)*

Field Name	Field Description
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.

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 on page 63
- show openflow statistics flows on page 81
- show openflow statistics interfaces on page 84
- show openflow statistics summary on page 91

List of Sample Output

show openflow statistics tables on page 94
show openflow statistics tables switch on page 94

Output Fields Table 27 on page 93 lists the output fields for the **show openflow statistics tables** command. Output fields are listed in the approximate order in which they appear.

Table 27: 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.

Table 27: show openflow statistics tables Output Fields *(continued)*

Field Name	Field Description
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 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 summary

Syntax **show openflow summary**

Release Information Command introduced in Junos OS Release 13.3.

Description Display summary information for OpenFlow including the number of configured switches, controllers, interfaces, and flows.

Options This command has no options.

Required Privilege Level admin

Related Documentation

- OpenFlow Operational Mode Commands on page 63
- show openflow statistics summary on page 91
- show openflow switch on page 96

Output Fields Table 28 on page 95 lists the output fields for the **show openflow summary** command. Output fields are listed in the approximate order in which they appear.

Table 28: 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.
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.

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 on page 63
- show openflow statistics tables on page 93
- show openflow summary on page 95

List of Sample Output
show openflow switch on page 97
show openflow switch OFswitch1 on page 97

Output Fields Table 29 on page 96 lists the output fields for the **show openflow switch** command. Output fields are listed in the approximate order in which they appear.

Table 29: show openflow switch Output Fields

Field Name	Field Description
Switch Name	User-configured identifier for the OpenFlow virtual switch.
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.

Table 29: show openflow switch Output Fields *(continued)*

Field Name	Field Description
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.
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 OFswitch1

```
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
```

PART 4

Index

Index

P

R

S

T

V