

53-1003257-03  
22 May 2015

# Brocade NetIron

---

## Software Defined Networking (SDN) Configuration Guide

Supporting Multi-Service IronWare R05.8.00b

**BROCADE** 

**© 2015, Brocade Communications Systems, Inc. All Rights Reserved.**

ADX, Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, HyperEdge, ICX, MLX, MyBrocade, OpenScript, The Effortless Network, VCS, VDX, Vplane, and Vyatta are registered trademarks, and Fabric Vision and vADX are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of others.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit <http://www.brocade.com/support/oscd>.

# Contents

---

<b>Preface.....</b>	<b>7</b>
Document conventions.....	7
Text formatting conventions.....	7
Command syntax conventions.....	7
Notes, cautions, and warnings.....	8
Brocade resources.....	9
Contacting Brocade Technical Support.....	9
Document feedback.....	10
<b>About This Document.....</b>	<b>11</b>
Audience.....	11
Supported hardware and software.....	11
Supported software.....	12
Notice to the reader.....	12
Related publications.....	13
How command information is presented in this guide.....	13
<b>OpenFlow.....</b>	<b>15</b>
Overview of OpenFlow.....	16
Flow table entries.....	18
OpenFlow actions .....	20
OpenFlow Controller.....	21
OpenFlow counters.....	21
Considerations and limitations for configuring OpenFlow.....	23
Brocade NetIron CER Series and Brocade NetIron CES Series devices.....	23
Brocade NetIron MLX Series and Brocade NetIron XMR Series devices.....	23
OpenFlow hybrid switch mode and OpenFlow hybrid port mode .....	24
Hybrid switch mode.....	24
Hybrid port mode.....	24
Rate limiting capabilities on OpenFlow enabled ports.....	27
VPLS support for VLANs on OpenFlow hybrid mode ports.....	28
Limitations.....	28
OpenFlow on individual LAG ports.....	32
Configuring OpenFlow.....	33
Enabling OpenFlow on devices.....	33
Connecting to an OpenFlow controller.....	36
Setting up SSL encryption for controller connections.....	36
Configuring multiple controller connections.....	37
Configuring the system parameters for OpenFlow.....	37
Configuring the default action.....	38
Displaying the OpenFlow status on the device.....	39
Displaying the OpenFlow status.....	40
Displaying the configured connections to controllers.....	41
Displaying the data path ID of the device.....	42
Displaying the OpenFlow flows.....	42
Purge-time for OpenFlow.....	44

Administrating OpenFlow.....	44
Clearing the OpenFlow statistics.....	44
Deleting the OpenFlow flows.....	44
Show tech.....	45
OpenFlow configuration considerations.....	45
Behavior of ports and devices.....	45
Removing an OpenFlow configuration from a device.....	46
<b>OpenFlow 1.3.....</b>	<b>47</b>
Overview of OpenFlow 1.3.....	47
OpenFlow IPv6 rule support.....	49
Flow table entries.....	61
OpenFlow instructions.....	62
OpenFlow actions.....	64
Multiple controller connections.....	64
Asynchronous configuration.....	65
Configuring source-interface for the controllers.....	66
Supported OpenFlow messages.....	67
Displaying the OpenFlow flows.....	68
Normal action.....	69
Supporting untagged VLAN on protected and configured unprotected VLAN.....	70
Group table.....	72
Scaling group numbers.....	72
Considerations and limitations for group tables.....	73
Supporting flow formats for group.....	74
Group events.....	75
QinQ.....	76
Flow validation checks.....	77
QinQ action.....	77
Enqueue.....	82
Configuring OpenFlow Enqueue.....	83
Limitations.....	83
Queue statistics.....	83
Displaying OpenFlow queues.....	84
Metering.....	85
Meter statistics.....	86
Limitations.....	87
Displaying OpenFlow meters.....	87
TTL support.....	88
Forwarding port and controller for matching ARP.....	89
sFlow support on OpenFlow ports.....	89
TLS1.2 support for NetIron devices.....	90
<b>SDN Commands.....</b>	<b>91</b>
show openflow.....	92
show openflow controller.....	94
openflow controller source-interface.....	95
show openflow interfaces.....	96
show openflow flows.....	98
openflow enable .....	100
clear openflow .....	101
show openflow groups.....	102
show openflow queues.....	103
show openflow meters.....	105
clear statistics openflow .....	107

openflow hello-reply disable..... 108



# Preface

---

- Document conventions..... 7
- Brocade resources..... 9
- Contacting Brocade Technical Support..... 9
- Document feedback..... 10

## Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

### Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
<b>bold text</b>	Identifies command names Identifies keywords and operands Identifies the names of user-manipulated GUI elements Identifies text to enter at the GUI
<i>italic text</i>	Identifies emphasis Identifies variables Identifies document titles
<code>Courier font</code>	Identifies CLI output Identifies command syntax examples

### Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
<b>bold text</b>	Identifies command names, keywords, and command options.
<i>italic text</i>	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, <b>--show</b> WWN.

Convention	Description
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x   y   z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x   y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member[member...]</i> .
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

## Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

---

### NOTE

A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

---

---

### ATTENTION

An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

---



### CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

---



### DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

---



## Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at [www.brocade.com](http://www.brocade.com). Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to [MyBrocade](#). You can register at no cost to obtain a user ID and password.

Release notes are available on [MyBrocade](#) under Product Downloads.

White papers, online demonstrations, and data sheets are available through the [Brocade website](#).

## Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

### Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to <http://www.brocade.com/services-support/index.html>.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
<p>Preferred method of contact for non-urgent issues:</p> <ul style="list-style-type: none"> <li>• <a href="#">My Cases</a> through MyBrocade</li> <li>• <a href="#">Software downloads</a> and licensing tools</li> <li>• <a href="#">Knowledge Base</a></li> </ul>	<p>Required for Sev 1-Critical and Sev 2-High issues:</p> <ul style="list-style-type: none"> <li>• Continental US: 1-800-752-8061</li> <li>• Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33)</li> <li>• For areas unable to access toll free number: +1-408-333-6061</li> <li>• <a href="#">Toll-free numbers</a> are available in many countries.</li> </ul>	<p><a href="mailto:support@brocade.com">support@brocade.com</a></p> <p>Please include:</p> <ul style="list-style-type: none"> <li>• Problem summary</li> <li>• Serial number</li> <li>• Installation details</li> <li>• Environment description</li> </ul>

### Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.

- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

## Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on [www.brocade.com](http://www.brocade.com).
- By sending your feedback to [documentation@brocade.com](mailto:documentation@brocade.com).

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

# About This Document

---

- Audience..... 11
- Supported hardware and software..... 11
- Notice to the reader..... 12
- Related publications..... 13
- How command information is presented in this guide..... 13

## Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Brocade device, you should be familiar with the following protocols if applicable to your network - IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

## Supported hardware and software

The following hardware platforms are supported by this release of this guide:

**TABLE 1** Supported devices

Brocade NetIron XMR Series	Brocade MLX Series	NetIron CES 2000 and NetIron CER 2000 Series
Brocade NetIron XMR 4000	Brocade MLX-4	Brocade NetIron CES 2024C
Brocade NetIron XMR 8000	Brocade MLX-8	Brocade NetIron CES 2024F
Brocade NetIron XMR 16000	Brocade MLX-16	Brocade NetIron CES 2048C
Brocade NetIron XMR 32000	Brocade MLX-32	Brocade NetIron CES 2048CX
	Brocade MLXe-4	Brocade NetIron CES 2048F
	Brocade MLXe-8	Brocade NetIron CES 2048FX
	Brocade MLXe-16	Brocade NetIron CER 2024C
	Brocade MLXe-32	Brocade NetIron CER-RT 2024C
		Brocade NetIron CER 2024F
		Brocade NetIron CER-RT 2024F
		Brocade NetIron CER 2048C
		Brocade NetIron CER-RT 2048C
		Brocade NetIron CER 2048CX
		Brocade NetIron CER-RT 2048CX
		Brocade NetIron CER 2048F
		Brocade NetIron CER-RT 2048F
		Brocade NetIron CER 2048FX
		Brocade NetIron CER-RT 2048FX

## Supported software

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the *Brocade NetIron R05.8.00b Release Notes*.

## Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Internet Explorer
Mozilla Corporation	Mozilla Firefox
Sun Microsystems	Java Runtime Environment

## Related publications

For the latest edition of these documents, which contain the most up-to-date information, see Documentation at <http://www.brocade.com/ethernetproducts>

- Brocade NetIron Administration Guide
- Brocade NetIron Security Configuration Guide
- Brocade NetIron Switching Configuration Guide
- Brocade NetIron Routing Configuration Guide
- Brocade NetIron Traffic Management Configuration Guide
- Brocade NetIron Multicast Configuration Guide
- Brocade NetIron Multiprotocol Label Switch (MPLS) Configuration Guide
- Brocade NetIron Software Defined Networking (SDN) Guide
- *Brocade MLX Series and NetIron Family YANG Guide*
- *Brocade MLX Series and NetIron XMR Series Diagnostic Reference*
- *Unified IP MIB Reference*
- *Brocade NetIron Software Upgrade Procedures for Brocade MLX Series and NetIron Family devices*
- *Brocade MLXe Series Installation Guide*
- *Brocade MLX Series and Brocade NetIron XMR Installation Guide*
- *Brocade NetIron CES 2000 Series and Brocade NetIron CER 2000 Series Hardware Installation Guide*

## How command information is presented in this guide

For all new content, command syntax and parameters are documented in a separate command reference section at the end of the publication.

In an effort to provide consistent command line interface (CLI) documentation for all products, Brocade is in the process of preparing standalone Command References for the IP platforms. This process involves separating command syntax and parameter descriptions from configuration tasks. Until this process is completed, command information is presented in two ways:

- For all new content included in this guide, the CLI is documented in separate command pages. The new command pages follow a standard format to present syntax, parameters, usage guidelines, examples, and command history. Command pages are compiled in alphabetical order in a separate command reference chapter at the end of the publication.
- Legacy content continues to include command syntax and parameter descriptions in the chapters where the features are documented.

If you do not find command syntax information embedded in a configuration task, refer to the command reference section at the end of this publication for information on CLI syntax and usage.

How command information is presented in this guide

# OpenFlow

- [Overview of OpenFlow](#)..... 16
- [Considerations and limitations for configuring OpenFlow](#)..... 23
- [OpenFlow hybrid switch mode and OpenFlow hybrid port mode](#) ..... 24
- [Configuring OpenFlow](#)..... 33
- [Administrating OpenFlow](#)..... 44
- [Show tech](#)..... 45
- [OpenFlow configuration considerations](#)..... 45

This table lists the individual Brocade NetIron devices and the OpenFlow features they support.

**TABLE 2** Supported Brocade OpenFlow features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 BASE package	Brocade NetIron CES Series 2000 ME_PREM package	Brocade NetIron CES Series 2000 L3_PREM package	Brocade NetIron CER Series 2000 Base package	Brocade NetIron CER Series 2000 Advanced Services package
OpenFlow v1.0.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OpenFlow enabled on per-port basis (Hybrid switch mode)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OpenFlow Layer 3 Hybrid Port Mode	Yes	Yes	No	No	No	No	No
Maximum number of flows supported per-port	4096	4096	4096	4096	4096	4096	4096
OpenFlow scaling	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Maximum number of flows supported per-system	128K	64K	4K	4K	4K	32K	32K
Support for hybrid port mode with protected VLANs and unprotected VLANs on VPLS endpoint ports	Yes	Yes	No	No	No	No	No

**TABLE 2** Supported Brocade OpenFlow features (Continued)

Features supported	Brocade Netron XMR Series Series	Brocade Netron MLX Series	Brocade Netron CES Series 2000 Series BASE package	Brocade Netron CES Series 2000 ME_PREM package	Brocade Netron CES Series 2000 L3_PREM package	Brocade Netron CER Series 2000 Base package	Brocade Netron CER Series 2000 Series Advanced Services package
Layer 2 OpenFlow match rules	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Layer 3 OpenFlow match rules	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Simultaneous Layer2 and Layer3 support for OpenFlow	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Match both source and destination MAC addresses	Yes	Yes	No	No	No	No	No
Support for passive mode on the device	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OpenFlow actions	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Maximum flow statistics	4096	4096	2048	2048	2048	2048	2048
CLI command to delete a single flow or all flows in the flow table	Yes	Yes	Yes	Yes	Yes	Yes	Yes

## Overview of OpenFlow

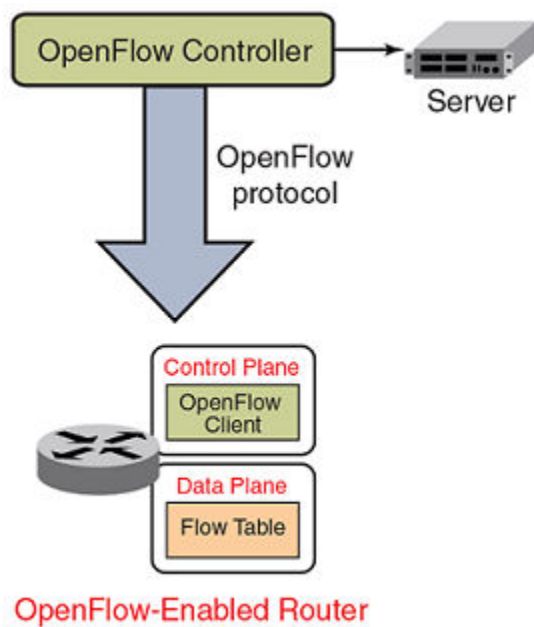
An OpenFlow-enabled router supports an OpenFlow Client (control plane software), which communicates with an OpenFlow Controller using the OpenFlow protocol. The OpenFlow Controller runs on a server or a server cluster. OpenFlow-enabled routers support the abstraction of a flow table, which is manipulated by the OpenFlow Controller. The flow table contains flow entries. Each flow entry represents a flow (that is, packets with a given MAC address, VLAN tag, IP address, or TCP/UDP port, and so on). The flow table is sorted by flow priority, which is defined by the OpenFlow Controller. The highest priority flows are at the top of the flow table.

Incoming packets on an OpenFlow-enabled port are matched (in order of priority) against the flow entries defined for that port by the OpenFlow Controller. If the packet matches a given flow entry, the

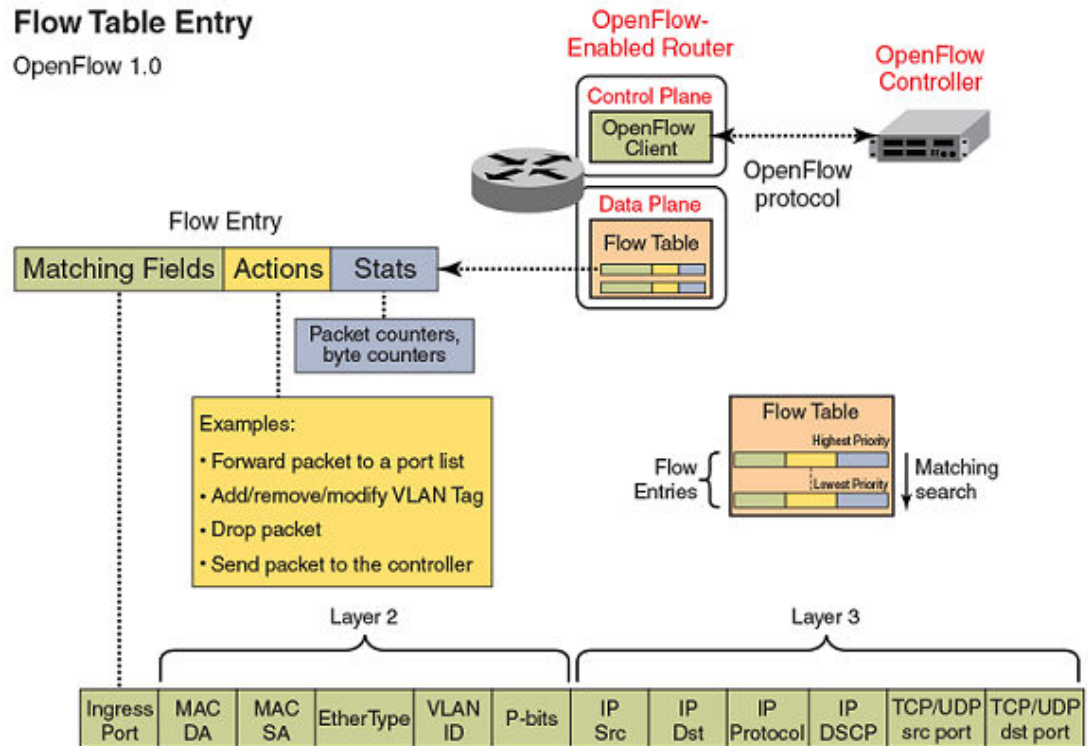


flow-matching process stops, and the set of actions defined for that flow entry are performed. Packets that don't match any flow entry are dropped by default. The Brocade implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to [Configuring the default action](#) on page 38.

**FIGURE 1** OpenFlow-enabled router



**FIGURE 2** OpenFlow flow table entries



## Flow table entries

The OpenFlow match rules in the following table are supported on Brocade devices for Flow table entries.

The implementation of OpenFlow supports three modes of operation when enabling OpenFlow on a port: Layer 2 mode, Layer 3 mode and Layer23 mode. Layer 2 mode supports OpenFlow matching rules based on the Layer 2 fields shown in [Overview of OpenFlow](#) on page 16, while Layer 3 mode supports the OpenFlow matching rules based on the Layer 3 fields. Layer23 mode supports the OpenFlow matching rules based on the Layer 2 and Layer 3 fields.

The Brocade NetIron MLX Series and Brocade NetIron XMR Series devices support enabling ports in either Layer 2 or Layer 3 mode. The Brocade NetIron CER Series and Brocade NetIron CES Series devices support Layer 2 mode by default (OpenFlow Layer 3 mode configuration on a port is currently not supported on these devices).

**TABLE 3** OpenFlow match rules

Match rule	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Port enabled for Layer 2 mode	Yes	Yes
Source port	Yes	Yes

**TABLE 3** OpenFlow match rules (Continued)

Match rule	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Source or destination MAC address	Yes These devices support either source or destination MAC address, or a combination of both source and destination MAC addresses as the match rule.	These devices support either source MAC or destination MAC address as the match rule, not both at the same time. This is a global option to be configured by the user.
Ether type	Yes	Yes
VLAN ID	Yes	Yes
VLAN priority	Yes	Yes
Untagged packets	Yes	Yes
<b>Port enabled for Layer 3 mode</b>	Yes	No
Ether type (Supports matching to Ether type value 0x88CC LLDP only; supports only 'send to controller' as the action)	Yes	No
Ether type (Supports matching to Ether type ARP value; supports only 'send to controller' as the action)	Yes	No
Source port	Yes	No
VLAN ID	Yes	No
VLAN priority	Yes	No
Source IP address	Yes	No
Destination IP address	Yes	No
Protocol type	Yes	No
IP TOS bits	Yes	No
TCP or UDP source port	Yes	No
<b>Port enabled for Layer23 mode</b>	Yes	Yes
Ether type (Supports matching to Ether type value 0x88CC LLDP only; supports only 'send to controller' as the action)	Yes	Yes

**TABLE 3** OpenFlow match rules (Continued)

Match rule	Brocade Netron MLX Series	Brocade Netron CER Series
	Brocade Netron XMR Series	Brocade Netron CES Series
Ether type (Supports matching to Ether type ARP value; supports only 'send to controller' as the action)	Yes	Yes
Source port	Yes	Yes
VLAN ID	Yes	Yes
VLAN priority	Yes	Yes
Source IP address	Yes	Yes
Destination IP address	Yes	Yes
Protocol type	Yes	Yes
IP TOS bits	Yes	Yes
TCP or UDP source port	Yes	Yes
TCP or UDP destination port	Yes	Yes

## OpenFlow actions

Each OpenFlow flow table entry contains the list of actions to be performed when a packet matches the flow entry. These actions are defined by the OpenFlow Controller.

Packets that do not match any flow entry are dropped by default. The Brocade implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to [Configuring the default action](#) on page 38.

Brocade devices support the actions listed in the following table .

**TABLE 4** OpenFlow actions supported on Brocade devices

OpenFlow action	Brocade Netron MLX Series	Brocade Netron CER Series
	Brocade Netron XMR Series	Brocade Netron CES Series
Forward a packet to a port or set of ports	Yes	Yes
Drop the packet	Yes	Yes
Add, modify, or remove VLAN ID or priority on a per destination port basis	Yes	Yes

**TABLE 4** OpenFlow actions supported on Brocade devices (Continued)

OpenFlow action	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Modify the IP DSCP (for a flow sending a copy of the packet to multiple destinations, the DSCP modification must be the same for all destinations. Modifying IP DSCP is only supported on ports enabled with Layer 3 mode.)	Yes	No
Modify the source or destination MAC address (for a flow sending a copy of the packet to multiple destinations, the MAC address modification must be the same for all destinations).	Yes	No
Send the packet to the controller (Packet In)	Yes	Yes
Receive the packet from the controller and send it to ports (Packet Out)	Yes	Yes

## OpenFlow Controller

Multiple controller connections can be used for redundancy purposes, such as when using a single controller with multiple addresses. Multiple controller connections can also be used to support active-standby controllers.

Regardless of the intended use of multiple controller connections, the Brocade device allows all the controller connections to concurrently manage the flow table. That is, flow entries in the flow table are not identified as belonging to any specific controller connection. In an active-standby controller deployment, controllers themselves must coordinate their actions and active-standby states. The Brocade device will respond to all connected controllers without distinction.

The Brocade device supports two types of controller connections (also called modes): active and passive. An active connection is one for which the Brocade device will initiate (seek) the TCP connection to a given OpenFlow Controller address. With a passive connection, the Brocade device will passively wait for the controller to initiate (seek) the TCP connection to the Brocade device. Active mode is commonly used with production controllers, while passive mode is commonly used for testing purposes in experimental environments. Optionally, a controller connection can also use SSL encryption.

## OpenFlow counters

Brocade NetIron MLX Series and Brocade NetIron XMR Series devices record the number of received packets and bytes on a per-flow basis. However, recording the number of received bytes on a per-flow basis is only supported on ports on the 8x10G or 100G cards. Brocade NetIron CER Series and Brocade NetIron CES Series devices record only the received packets on a per-flow basis—these devices do not record the number of bytes per flow.

The following per port counters are available in the flow table:

**TABLE 5** OpenFlow counters supported on devices

Counter	Description	Brocade Netron MLX Series	Brocade Netron CER Series
		Brocade Netron XMR Series	Brocade Netron CES Series
Received packets	Number of packets received on the port	Yes	Yes
Transmitted packets	Number of packets transmitted from the port	Yes	Yes
Received bytes	Number of bytes received on the port	Yes	No
Transmitted bytes	Number of bytes transmitted from the port	Yes	Yes
Receive drops	Number of received packets dropped on the port because the packets did not match any rules	Yes	Yes
Transmit drops	Number of transmit packets dropped on the egress port	Yes	Yes
Receive errors	Number of errors detected on the port on received packets	Yes	Yes
Transmit errors	Number of errors detected on the port on transmitted packets	Yes	Yes
Receive frame alignment errors	Number of frame alignment errors detected on packets received on the port	Yes	No
Receive overrun errors	Number of packets that caused overrun in the receive buffer on the port	Yes	No
Receive CRC errors	Number of packets received on the port that had CRC errors	Yes	No
Collisions	Number of collisions recorded on the port	Yes	No

The following table lists the per-flow counters available:

**TABLE 6** Per-flow OpenFlow counters supported

Counter	Description	Brocade Netron MLX Series	Brocade Netron CER Series Brocade Netron CES Series
		Brocade Netron XMR Series	
Received packets	Number of packets received per flow	Yes	Yes
Received bytes	Number of bytes received per flow	Only on 8x10G or 100G cards	No

## Considerations and limitations for configuring OpenFlow

Consider the following points when you configure OpenFlow on devices:

- OpenFlow must be enabled globally on the device before you can enable interfaces for OpenFlow.
- You must explicitly enable or disable OpenFlow on each interface using the CLI commands. You cannot use a range of ports to enable OpenFlow on the interface.
- Before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- Spanning Tree Protocol and other Layer 2 or Layer 3 protocols are not supported on OpenFlow-enabled ports.
- OpenFlow supports up to three concurrent sessions with a maximum of two concurrent SSL sessions.
- OpenFlow supports up to 3000 configured flows are supported if all the flows are with a wildcard for the incoming port.
- On hybrid OpenFlow ports, Layer 2 unicast and multicast packets will be flooded in the VLAN for protected VLANs and for unprotected VLANs in absence of flows.
- Local and normal actions defined by the OpenFlow 1.0 protocol are not supported.
- After a reboot, a delay of 16 seconds for Brocade NetIron CER Series and Brocade NetIron CES Series devices, and 180 seconds for Brocade NetIron MLX Series and Brocade NetIron XMR Series devices is added before any flows can be pushed. This is the default delay time to allow all modules and ports to come up.
- 8x10GbE and 2x100GbE interface modules support all generic Ethertype value in Layer 2 mode. All other interface modules support Ethertype 0x88CC (LLDP) only in Layer 2 mode.
- OpenFlow is an ingress feature. The local device will generate protocol messages (such as PIM, OSPF) on OpenFlow enabled-ports, if configured, but return control packets will be filtered or denied by OpenFlow default rule. Because of this limitation, the PIM neighbor (if configured) will come up on the peer, and multicast traffic will hit the OpenFlow interface in all PIM DMs, and in a PIM SM scenario in which the OpenFlow port connects to an IGMP snooping-enabled LAN that has the multicast source connected.
- On OpenFlow-enabled ports, packets that do not match any flow entry are dropped by default. The Brocade implementation supports an option to send such packets to the OpenFlow Controller. Refer to [Configuring the default action](#) on page 38.

### Brocade NetIron CER Series and Brocade NetIron CES Series devices

- Enabling ports for Layer 3 flows is not supported.
- Either Source MAC address or Destination MAC address can be used as a matching rule. The choice is a device-level configuration that you specify at the time of enabling OpenFlow on the device.
- Statistics are available only for up to 2048 flows on a first-come-first-served basis.

### Brocade NetIron MLX Series and Brocade NetIron XMR Series devices

- Even though modification of the source and destination MAC address fields of a packet is supported, modification on a per-destination port basis for multi-destination flows is not supported.

In other words, sending a packet to multiple ports and having the MAC addresses modified on a per-destination port basis is not supported. However, sending a packet to multiple ports and having the MAC addresses modified in the same way for all destination ports is supported. Similar restriction applies to modifications of the DSCP field of packets.

- Modification of VLAN tag (such as adding, removing, modifying, or modifying the PCP bits of a tag) is supported on a per-destination port basis for tagged packets. A flow may specify sending a copy of a

packet to multiple destination ports and have each copy of the packet tagged with a different VLAN ID.

At the same time, if the rule is to send a tagged packet to multiple outgoing ports with different VLAN tags, and if the action for one of the outgoing ports is to send the packet as untagged, the packet is still tagged with the VLAN ID of 0 on that port. For egress ports with the action specifying modified VLAN tag, the specified VLAN tag itself is added.

- An action to modify the VLAN priority of an untagged frame will result in a VLAN tag being added to the frame. If the action does not specify a VLAN ID value, the VLAN ID will be set to 0.
- The action to modify IP DSCP is only supported for flows on ports enabled for Layer 3 mode.
- Matching a VLAN ID on a Layer 3 mode port is only supported for packets with an IP payload.

## OpenFlow hybrid switch mode and OpenFlow hybrid port mode

### Hybrid switch mode

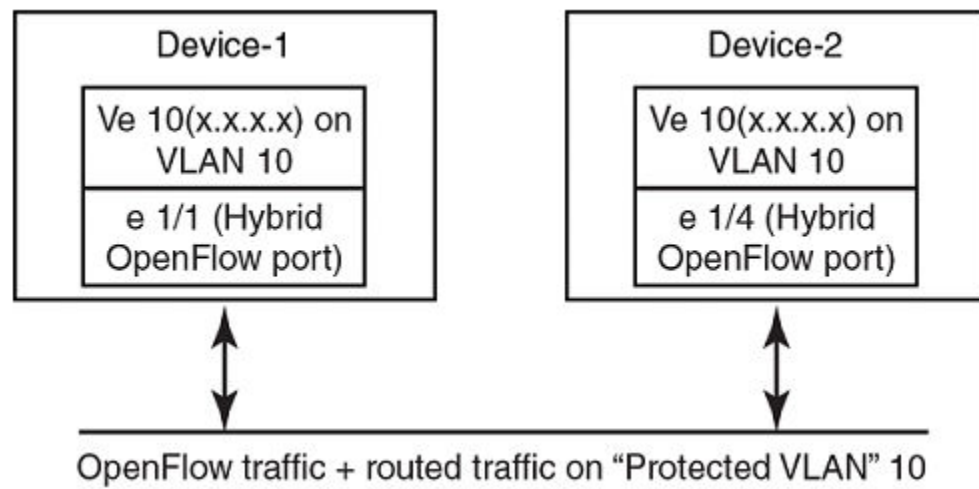
The Brocade device supports enabling OpenFlow on a per-port basis, so you can choose which ports of the device will be controlled by the OpenFlow feature. Non-OpenFlow-enabled ports continue to support existing features of the device, such as MPLS, VPLS, IPv4 or IPv6 routing for Layer 2 switching.

### Hybrid port mode

OpenFlow hybrid-enabled ports support both OpenFlow traffic forwarding and normal routing traffic forwarding. OpenFlow hybrid-enabled ports support "protected VLANs" and "unprotected VLANs". Protected VLANs are not subject to defined OpenFlow flows on the OpenFlow hybrid-enabled ports. OpenFlow flows on a hybrid-enabled port will not match any traffic on protected VLANs. Unprotected VLANs are subject to defined OpenFlow flows on the OpenFlow hybrid-enabled port. OpenFlow flows on a hybrid-enabled port are allowed to match on the traffic of unprotected VLANs.

[Figure 3](#) shows a topology in which port 1/1 on Device-1 and port 1/4 on Device-2 are hybrid-enabled OpenFlow ports with VLAN 10 as a configured protected VLAN. By configuring a virtual ethernet (VE) interface on a protected VLAN 10 and assigning an address to route the traffic of the nodes, you are able to send protected VLAN traffic between the nodes and route the traffic as per the VE interface. Traffic flowing on other VEs created on top of other VLANs (the unprotected VLANs ) is treated as unprotected VLAN traffic and is subject to OpenFlow rules lookup. OpenFlow traffic can be forwarded through this port.



**FIGURE 3** OpenFlow hybrid port mode topology

### ***OpenFlow hybrid port mode operation***

Consider Device-1 in [Hybrid port mode](#) on page 24. Ingress traffic on VLAN 10 on hybrid port 1/1 will be processed for IPv4 and IPv6 unicast routing. Traffic on other VLANs will be processed against OpenFlow flows on port 1/1 and switched accordingly. A preconfigured number of protected VLANs can be supported for normal routing. The Spanning Tree Protocols (STP) state of these routing VLANs will be set to forwarding, as the Layer 2 protocol is not supported.

### ***Configuring OpenFlow hybrid port mode***

1. Enable OpenFlow at the global configuration level.
2. Configure OpenFlow controller configurations.
3. Configure the system maximum configuration for the maximum OpenFlow entries. (The default is 0.)
4. Configure the maximum OpenFlow flow-protected VLAN entries. (The default is 0.)

---

#### **NOTE**

System reload is required once you change the system maximum values.

---

5. Configure the maximum OpenFlow unprotected VLAN entries. (The default is 0.)
6. Configure protected VLANs on the port. A maximum of 40 protected VLANs can be configured on an OpenFlow port.
7. Enable OpenFlow hybrid port mode on the desired interfaces.
8. Configure a VE for the interface by specifying the protected VLAN and add routing entries.

### ***Feature information***

- Switchover and HLOS are not supported. When the active management processor (MP) goes down, communication with the controller is brought down and the flow tables on the MP and all line processors (LP) are cleared. The connection with the controller is re-established after switchover.
- When LP is reset, the flow table on the LP is restored once the LP comes up and flows specific to that LP are maintained in the MP.

- When an OpenFlow-enabled port goes up or down, no rules are removed. The addition or deletion of rules depends solely on the controller.
- 4K OpenFlow entries are supported.
- Up to 2K protected VLANs per system are supported.

### ***Capabilities and prerequisites***

The following are current capabilities and prerequisites of OpenFlow hybrid port mode:

- IPv4 and IPv6 unicast routing are supported on OpenFlow protected and unprotected VLANs.
- Packets tagged with a protected VLAN ID will be forwarded by IPv4 and IPv6 unicast routing, if IPv4 or IPv6 routing is configured on that VLAN. If IPv4 or IPv6 routing is not configured on that VLAN, such packets will be dropped.
- Packets tagged with an unprotected VLAN ID will be subject first to OpenFlow flows. If there is a match on an OpenFlow flow, the packet will be forwarded according to the flow actions. No further IPv4 or IPv6 routing is supported for packets that are forwarded by OpenFlow flows. If there is no match on any OpenFlow flow, the packet will be forwarded by IPv4 or IPv6 unicast routing, if IPv4 or IPv6 routing is configured on the VLAN. If IPv4 or IPv6 routing is not configured on the VLAN, those packets are either dropped or sent to the controller, per the OpenFlow configuration.
- Layer 2 or L2VPN forwarding is not supported on ports in hybrid-enabled ports because MAC learning is disabled on these ports.
- A port can be enabled for OpenFlow hybrid port mode only if the port is untagged in the default VLAN.
- Ports in OpenFlow hybrid port mode cannot be added as untagged ports to regular VLANs or L2VPN because this can cause a problem with topology discovery.
- As routing is enabled on a port in OpenFlow hybrid port mode, OpenFlow traffic or unprotected VLAN traffic sent with the destination MAC address as the port's MAC address and matching IP route entries on the port can potentially find the VLAN and MAC address modified unless the OpenFlow rules explicitly set the VLAN and destination MAC address in the outgoing packet.
- Inbound normal ACL configuration is not supported on the port in hybrid port mode.
- Any port with the default VLAN not equal to the system default VLAN ID cannot be enabled for hybrid port mode.
- Policy-based routing (PBR) is not supported.
- Protected VLAN traffic that does not have matching IP route entries will be dropped.
- Multiple interfaces cannot be part of a VE interface created on a port in OpenFlow hybrid port mode with a protected VLAN.
- The BGP, OSPF, and IS-IS protocols are supported on protected VLANs:

---

#### **NOTE**

Layer 2 or L2VPN, VRF are not supported.

---

- When protected VLANs are configured but the port is not part of the VLAN, the traffic coming on the port with the protected VLAN will be dropped.
- Port in hybrid-enabled OpenFlow doesn't support MPLS running on the same port.
- Link aggregation is not supported.

### ***Enabling OpenFlow hybrid port mode***

Use the **openflow enable** command to enable or disable OpenFlow hybrid port mode on the port and the port becomes a normal port on an interface. The **no** form of the command disables the OpenFlow hybrid port mode on the port and the port becomes a normal port.

```
config-if-e10000-2/5)# openflow enable layer2 hybrid-mode
```

**Syntax:** [no] openflow enable layer2 | layer3 | layer23 [hybrid-mode ]

### ***Adding or Deleting protected VLANs***

Use **openflow protected -vlangs** to add or delete protected VLANs on a hybrid port-mode interface. The **no** form of the command deletes the configured protected VLANs from the hybrid-enabled port.

```
device(config-if-e10000-2/5)# openflow protected-vlangs 10
```

**Syntax:** [no] openflow protected-vlangs id1 id2...idn

VLANs can be configured individually.

---

#### **NOTE**

A VLAN range is not specified for this command .

---

### ***Displaying OpenFlow configuration***

The **show openflow** command displays the OpenFlow configuration.

```
device(config)# show openflow
Number of Controllers:      2
Controller 1:
Connection Mode:           passive, TCP,
Listening Address:         0.0.0.0
Connection Port:           6633
Connection Status:
SSL Connection              :False
No TCP connection found.
Controller 2:
Connection Mode:           active, TCP,
Controller Address:        10.20.101.199
Connection Port:          23
Connection Status:
Local IP address:port <-> Remote IP address:port TCP state   RcvQue  RxBuffe  SendQue
TxBuffe
10.20.178.73      8807      10.20.101.199  23      ESTABLISHED  0        0        0
0
SSL Connection              :False
Match Capabilty:
L2: Port, Source MAC, Destination MAC, Ether type, Vlan, Vlan PCP
L3: Port, Vlan, Vlan PCP, Source IP, Destination IP, IP Protocol, IP TOS, IP Src
Port, IP Dst Port
Normal Openflow Enabled Ports:      e2/1
Hybrid Mode ports      Protected Vlan-IDs
e4/1                    (100,101,102,103)
e7/2                    (200)
Default action: drop
Maximum number of flows allowed: 4096
Maximum number of Protected Vlangs allowed: 2048
```

## **Rate limiting capabilities on OpenFlow enabled ports**

Rate-limiting support on OpenFlow enabled ports:

- OpenFlow ports (non-hybrid port mode ports): Normal CLI configuration of port-based rate limiting is supported. Per VLAN rate limiting not supported.
- You must set the rate control for certain protocols at the global configuration level.

---

**NOTE**

Rate control for certain protocols, such as ARP, will be based on the global configuration. For a very large burst of ARP traffic on a port, which may not be a normal scenario but could happen, the system could become unresponsive.

---

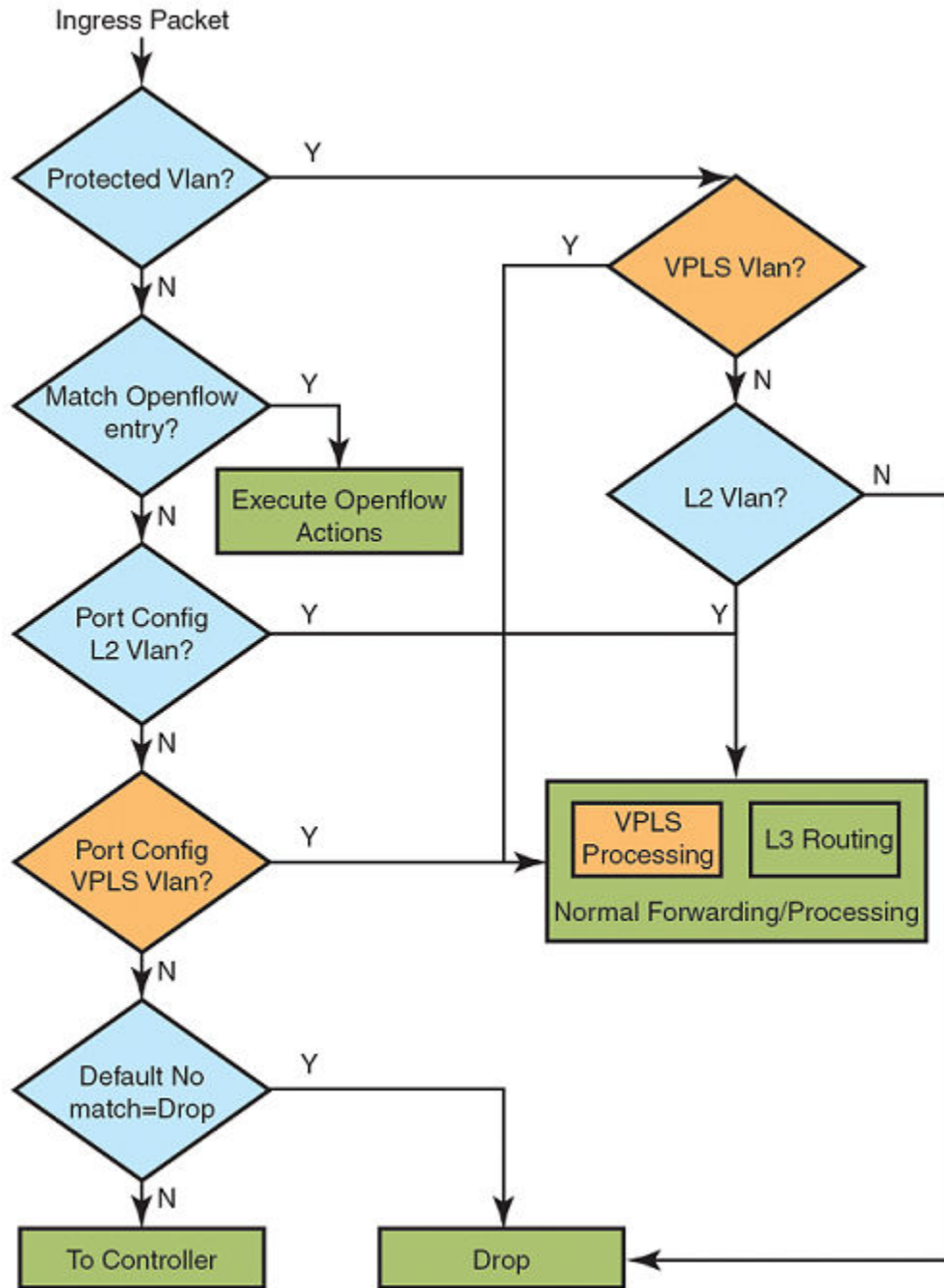
## VPLS support for VLANs on OpenFlow hybrid mode ports

This feature supports VPLS switching or routing for protected VLANs and for configured unprotected VLANs on OpenFlow hybrid mode ports. It is supported on Layer 2, Layer 3 and Layer23 OpenFlow hybrid mode ports.

### Limitations

- VPLS VLANs cannot be configured as Untagged VLANs on OpenFlow hybrid mode ports.
- OpenFlow unprotected VLANs for system maximum is limited to 4K. This number is utilized by both Layer 2 VLANs as well as VPLS VLANs.
- Layer 2 switching is not supported on OpenFlow hybrid mode ports.
- Dual tagged VPLS instances are not supported on OpenFlow hybrid mode ports.
- IGMP snooping is not supported on configured unprotected VLANs.
- If a VPLS instance is configured on an OpenFlow Layer 3 hybrid port and this VPLS VLAN is not protected (unprotected VLAN) then, in case of an incoming Layer 3 traffic (having same VLAN) hits a flow rule, it will not forward the traffic appropriately according to the flow rule.

**FIGURE 4** Packet flow diagram



**Sample configurations**

### ***VPLS support to configure OpenFlow hybrid mode port***

For VPLS instance, you can configure a port as an OpenFlow hybrid mode port by executing these commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
```

Check for global system-max unprotected VLAN number, while configuring a port as a hybrid port. If the system-max unprotected VLAN number exceeds the maximum permissible, the port is rejected from being configured as a hybrid mode port. Otherwise VPLS VLAN becomes the configured unprotected VLAN on the port.

### ***VPLS support on configured protected VLANs***

These are the steps to configure VLAN as an OpenFlow protected VLAN.

- Enable OpenFlow globally and configure **system-max** for the OpenFlow entries.
- Configure the OpenFlow Controller.
- Configure **system-max** for Protected VLAN entries.
- Enable OpenFlow hybrid on port.
- Configure protected VLAN for port.
- Configure VPLS instance on the port using the same configured protected VLAN.

To configure VLAN as an OpenFlow protected VLAN, execute the following commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
```

---

#### **NOTE**

The maximum protected VLAN number is 40 on the port. When it exceeds, you get an error message , and the VLAN will not be configured as a protected VLAN.

---

---

#### **NOTE**

If **system-max** protected VLANs exceeds 2K, then you get an error message , and the system will not configure the VLAN as protected VLAN.

---

### ***VPLS instance to enable OpenFlow hybrid mode port on protected VLAN***

Once protected VLANs have been configured, to enable the OpenFlow hybrid mode port, use the following commands to create a VPLS instance.

```
device(config)#
device(config)#router mpls
```

```

device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode

```

Since this protected VLAN has become part of VPLS VLAN on this port, VPLS switching on this protected VLAN will be supported.

### ***Deleting VPLS instances from protected VLAN***

To remove VPLS instance for the protected VLAN from the OpenFlow hybrid mode port, use this sequence of commands.

```

device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config)#
device(config)#router mpls
device(config)#vpls v1 100
device(config)#vlan 100
device(config)#no tag e 2/8

```

Now this port is not a part of VPLS instance. The protected VLAN becomes unconfigured protected VLAN and It will drop.

---

#### **NOTE**

When removing the protected VLAN configuration, do not exceed the **system-max** unprotected VLAN number. The command will be rejected and you get an error message.

---

After removing the protected VLAN configuration, this VPLS VLAN becomes configured as an unprotected VLAN. Now, It will do VPLS switching in absence of matching flow.

### **Deleting VPLS instances from unprotected VLAN**

To remove OpenFlow hybrid mode from a port with unconfigured protected VLAN and a VPLS instance, execute the following commands.

```

device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config)#router mpls
device(config)#vpls v1 100
device(config)#vlan 100
device(config)#no tag e 2/8

```

Now that, the port has become a normal port and VPLS instance is configured on the port, It will do VPLS processing for that VLAN.

To remove OpenFlow hybrid mode from a port with configured unprotected VLAN and a VPLS instance, execute the following commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v1-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config-if-e10000-2/8)#no openflow enable layer2 hybrid-mode
```

## OpenFlow on individual LAG ports

This feature is to enable OpenFlow hybrid on LAG ports such that non OpenFlow traffic forwarding is also supported on a LAG port. The controller can configure OpenFlow and traditional flows on individual LAG ports.

These are the limitations, when configuring LAG ports.

- All LAG ports need to behave the same way for non OpenFlow traffic.
- You can configure OpenFlow only on the primary LAG port. Secondary LAG ports will follow the same configuration as the primary LAG port.
- You cannot configure or delete OpenFlow on secondary LAG ports directly.
- When you look at the running configuration, the secondary ports will not be shown even though the secondary LAG ports have OpenFlow configuration taken from the primary port.
- Secondary LAG ports will take Protected VLANs configuration also from primary LAG port.
- Openflow enable or disable will take effect on secondary LAG port only when LAG is deployed.
- While LAG is deployed. LAG ports should not have any OpenFlow configuration. If they do, the LAG deploy will fail. Similarly, while LAG is undeployed, LAG ports should not have any OpenFlow configuration.

---

### NOTE

This feature is applicable for Brocade NetIron MLX Series and Brocade NetIron XMR Series

---

Example: Before enabling OpenFlow on primary LAG port, Ports 1/3, 1/4 and 1/5 are LAG ports and 1/3 is the primary port. OpenFlow is enabled on 1/1 and 1/2.

```
device#show openflow interfaces
Total number of Openflow interfaces: 2

Port  Link      Speed Tag MAC              OF-portid  Name      Mode
1/1   Up          1G   Yes 000c.dbf5.bd00  1         Layer2
1/2   Up          1G   Yes 000c.dbf5.bd01  2         Layer2
```



After enabling OpenFlow on primary LAG port, enable OpenFlow on LAG primary port 1/3.

```
device (config-if-e1000-1/3)#openflow enable layer3 hybrid
device (config-if-e1000-1/3)#show openflow interfaces
Total number of Openflow interfaces: 5
```

Port	Link	Speed	Tag	MAC	OF-portid	Name	Mode
1/1	Up	1G	Yes	000c.dbf5.bd00	1		Layer2
1/2	Up	1G	Yes	000c.dbf5.bd01	2		Layer2
1/3	Up	1G	Yes	000c.dbf5.bd01	3		Hybrid-Layer3
1/4	Up	1G	Yes	000c.dbf5.bd01	4		Hybrid-Layer3
1/5	Up	1G	Yes	000c.dbf5.bd01	5		Hybrid-Layer3

To disable OpenFlow, delete OpenFlow configuration from the LAG primary port. This will remove OpenFlow configuration from all secondary ports.

You must do the following to configure OpenFlow.

- Deploy LAG
- Configure OpenFlow (and protected VLANS) on primary LAG port
- Controller can configure flows on individual LAG ports
- Unconfigure OpenFlow (and protected VLANS) on primary LAG port
- Undeploy LAG

OpenFlow handles these LAG events.

- Deploy LAG: LAG ports should not have OpenFlow configuration at the time of LAG deploy. The system will check for OpenFlow configuration and reject LAG deploy if any OpenFlow configuration exists.
- Undeploy LAG: LAG ports should not have OpenFlow configuration at the time of LAG undeploy. The system will check for OpenFlow configuration and reject LAG undeploy if any OpenFlow configuration exists on the primary LAG port.
- Enabling or disabling of OpenFlow on deployed primary LAG port: OpenFlow enable or disable done on deployed LAG primary port should be applied to all secondary LAG ports. If any of the LAG port has flows, OpenFlow disable will fail. Similarly protected VLAN configuration done on primary LAG port should be applied to all secondary LAG ports.
- Addition or deletion of secondary port to deployed LAG: When you add secondary LAG port, OpenFlow configuration on primary port will be applied to newly added port. Similarly, when you delete secondary port from LAG, OpenFlow configuration on that port will be deleted.

## Configuring OpenFlow

You can enable OpenFlow on an interface with Layer23 option in order to support Layer 2 and Layer 3 flows on that interface. Layer23 option supports the OpenFlow hybrid port mode also. Configured with the Layer23 option, the controller can configure flows with Layer 2 and Layer 3 parameters together. A flow can contain the following fields, Ingress port, MAC DA, MAC SA, Ethertype, VLAN ID , P-bits, IP Src, IP Dst, IP protocol, and IP DSCP.

By default, OpenFlow is disabled on Brocade devices. You must first enable OpenFlow on the device before you can configure the parameters on the device.

### Enabling OpenFlow on devices

After you enable OpenFlow on the device, you can enable OpenFlow on specific interfaces and configure additional OpenFlow parameters.

### ***Enabling OpenFlow on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices***

Enter the following command:

```
device(config)# openflow enable ofv100
```

**Syntax: [no] openflow enable ofv100**

The **ofv100** keyword specifies the OpenFlow protocol version supported.

Use the **no** form of the command to disable OpenFlow feature on the device.

---

#### **NOTE**

You must disable OpenFlow on all interfaces individually before you can disable OpenFlow globally on the device.

---

### ***Enabling OpenFlow on Brocade NetIron CER Series and Brocade NetIron CES Series devices***

You can optionally specify the MAC address match rule capability as either source MAC or destination MAC address. Default is destination MAC address. On these devices, you cannot change the MAC address match option dynamically. You must first disable the current mode and then enable the new option. Changing the MAC address match option will clear all existing OpenFlow configuration and OpenFlow flow table content.

Enter the following command:

```
device(config)# openflow enable ofv100
```

**Syntax: [no] openflow enable ofv100 src-mac**

The **src-mac** keyword specifies the source MAC address matching option.

Use the **no** form of the command to disable OpenFlow feature on the device.

---

#### **NOTE**

You must disable OpenFlow on all interfaces individually before you can disable OpenFlow globally on the device.

---

### ***Enabling OpenFlow on a specified interface***

After you have enabled the OpenFlow feature on the device, you can enable OpenFlow on specific interfaces.

---

#### **NOTE**

You can enable OpenFlow on an interface only after you have enabled OpenFlow globally on the device. In addition, you must use individual CLI commands to enable OpenFlow on each interface. You cannot specify a range of ports when enabling OpenFlow on them.

---

---

#### **NOTE**

Configuration of an OpenFlow hybrid port is not supported, if the port is already configured as a member of an MCT VLAN.

---

### On Brocade NetIron CER Series and Brocade NetIron CES Series devices:

Enter the following command:

```
device(config-if-e1000-1/1)# openflow enable
```

**Syntax:** [no] openflow enable *layer2* | *layer23* [ *hybrid-mode* ]

Use the **no** version of the command to disable OpenFlow on the interface. By default, the port is enabled for Layer 2 matching mode, since Brocade NetIron CER Series and Brocade NetIron CES Series devices currently do not support Layer 3 matching mode. Layer23 can be enabled with hybrid mode. If hybrid mode is enabled, that interface will support protected and unprotected VLANs similar to Layer 2 hybrid and Layer 3 hybrid.

### On Brocade MLX Series and Brocade NetIron XMR Series devices

Enter the following command:

```
device(config-if-e1000-1/1)# openflow enable layer2
```

Or

```
device(config-if-e1000-1/1)# openflow enable layer3  
device(config-if-e1000-1/1)# openflow enable layer23
```

**Syntax:** [no] openflow enable [ *layer2* | *layer3* | *layer23* [ *hybrid-mode* ] ]

You can specify Layer 2 or Layer 3 or both layers in hybrid mode as Layer23 matching mode to be supported on the interface. By default, interfaces on these devices support Layer 2 matching mode. If you enable Layer 2 matching mode on the specified interface, only Layer 2 matching fields are supported on that interface.

### ***Flow validation***

The following validations are required before programming flows on a Layer23 port:

- When IP fields exist in rule, then the ETH\_TYPE must be 0x800.
- IPv6 rules are supported on the Layer23 port. (But IPv6 ETH\_TYPE without IPv6 parameters is supported.)

### ***Flow action***

OpenFlow actions are not changed for Layer23 support. All actions currently supporting Layer 2 or Layer 3 flows will continue to be supported. Actions currently supported are listed separately for different devices.

### On Brocade NetIron MLX Series and Brocade NetIron XMR Series devices:

When a matching flow entry is found, a set of actions can be applied for processing the packet. The system supports the following actions:

- Forward a packet to a port.
- Forward a packet to a set of ports.
- Forward a packet to a controller.
- Forward a packet received from a controller to a port or set of ports.
- Drop the packet.

- Keep, add, modify, or remove the VLAN ID or the VLAN priority. Modifying the VLAN ID per port is also supported (each destination port can send a packet with a different VLAN ID for the same matching rule).
- Modify the source MAC address and the destination MAC address for both Layer 2 and Layer 3 IPv4 flows.

### On Brocade NetIron CER Series and Brocade NetIron CES Series devices:

When a matching flow entry is found, a set of actions can be applied for processing the packet. The system supports the following actions:

- Forward a packet to a port.
- Forward a packet to a set of ports.
- Forward a packet to a controller.
- Forward a packet received from a controller to a port or set of ports.
- Drop the packet.
- Keep, add, modify, or remove the VLAN ID or the VLAN priority. Modifying the VLAN ID per port is also supported (each destination port can send a packet with a different VLAN ID for the same matching rule).

## Connecting to an OpenFlow controller

To connect to an OpenFlow controller in active mode, enter the following command:

```
device(config)# openflow controller ip-address 10.2.3.4
```

**Syntax:** **[no] openflow controller ip-address ip-address [ no-ssl ] [ port port ]**

The **ip-address** variable specifies the IP address of the OpenFlow Controller. By default, the connection with the Controller uses SSL encryption, but you can optionally disable SSL encryption using the **no-ssl** keyword. By default, the OpenFlow connection uses TCP port 6633, but you can specify another port using the **port** parameter.

Use the **no** form of the command to remove the specified OpenFlow Controller connection.

To connect to an OpenFlow controller in the passive mode, enter the following command:

```
device(config)# openflow controller passive no-ssl
```

**Syntax:** **[no] openflow controller passive no-ssl [ ip-address ip-address ] [ port port ]**

You can optionally specify the TCP port to be used for the connection. By default, the device accepts the connection from a controller with any IP address. However, you can provide an IP address to limit which controller can connect to the device.

Use the **no** form of the command to remove a passive connection. Passive mode connections are intended for testing environments and not recommended for production environments.

## Setting up SSL encryption for controller connections

By default, a connection to the controller uses SSL encryption. To set up SSL encryption, copy the SSL certificate and SSL client private key from the remote machine where you generated these into the device's flash using the following commands:

```
device(config)# copy tftp flash <remote ip> <remote file> client-certificate
device(config)# copy tftp flash <remote ip> <remote file> client-private-key
```

**Syntax:** `copy tftp flash remoteip | remotefile client-certificate`

**Syntax:** `copy tftp flash remote ip | remotefile client-private-key`

The **remote ip** variable specifies the IP address of the remote machine from which the SSL client certificate is being copied.

---

#### NOTE

SSL is not supported on passive controller connections.

---

The **remote file** variable specifies the file name of the client certificate in the first command, and the client private key in the second command.

For each controller, you must enter both the commands. The device can store up to three SSL certificates and client private keys. If you remove a controller connection, you will need to delete the SSL certificates and client private keys from the device's flash memory using the monitor mode commands.

### *Disabling an SSL client*

You can disable the SSL client within the device using the following command:

```
device# ip ssl client disable
```

**Syntax:** `[no] ip ssl client disable`

After you disable an SSL client in the device, the corresponding controller connection that used SSL encryption will fail. However, you can re-enable the controller connection by removing the SSL encryption option from the controller connection. Use the **no-ssl** option in the **openflow controller ip-address ip-address** command to disable SSL encryption in the connection.

To re-enable SSL client in the device, use the **no** form of **ip ssl client disable** command.

## Configuring multiple controller connections

Brocade devices support up to three controller connections. You can configure these connections with active or passive modes, in any combination, such as all active, all passive, or some active and some passive. Each connection requires its own separate command. You can remove any of the connections using the **no** form of the **openflow controller ip-address** command. The following example shows how you configure three connections.

```
device(config)# openflow controller ip-address 10.2.3.4 no-ssl port 6635
device(config)# openflow controller ip-address 10.2.3.5 no-ssl
device(config)# openflow controller passive no-ssl ip-address 10.2.3.6
```

## Configuring the system parameters for OpenFlow

You can specify the limit for OpenFlow flow table entries in the flow table using the following command:

```
device(config)# system-max openflow-flow-entries 304
```

**Syntax:** `system-max openflow-flow-entries limit`

Use the limit variable to specify the maximum number of flow table entries. The range is from 0 through 4096. The default number of flow table entries is zero.

---

**NOTE**

This command is not available on the Brocade NetIron CER Series and Brocade NetIron CES Series devices. The default number of flows table entries is 4096 on these devices.

---

### ***Setting the system maximum***

The **system-max openflow-pvlan-entries** command sets the CAM size of OpenFlow protected VLAN entries for the device. By default, this value is set to 0.

```
device(config)# system-max openflow-pvlan-entries 2000
```

**Syntax:** **system-max openflow-pvlan-entries** *value*

The *value* variable represents the number of port and protected VLAN combination entries that can be configured in the system. The range is from 0 through 2048. After using this command, you must reload the system.

The **system-max openflow-unprotectedvlan-entries** command sets the CAM size of OpenFlow unprotected VLAN entries for the device. By default, this value is set to 0.

```
device(config)# system-max openflow-unprotectedvlan-entries 1000
```

**Syntax:** **system-max openflow-unprotectedvlan-entries** *value*

The *value* variable represents the number of port and unprotected VLAN combination entries that can be configured in the system. The range is from 0 through 4096. After using this command, you must reload the system.

A new command has been added for Layer 2, Layer 3 and Layer23 flows.

**Syntax:** **system-max np-openflow-entries layer2or3 | layer23IPv4** *value slot [ i j k | i to z | all ]*

The following parameters are available for this command:

- layer2or3 -both Layer 2 flow or Layer 3 flow entries
- Layer23IPv4 - Layer23 including Layer 2 and IPv4 flow entries

The above layer2or3 or layer23IPv4 options are optional (at least one of them have to be specified).

- Slot number can be any of the valid slot number in the device. For slots, you can provide "all", "slot 1 to 2" and individual slot options.
- Slot number mentioned in a single command need not be of same card type.
- If any of the option layer2or3 or layer23IPv4 is not applicable for a particular card on a slot, it is ignored.
- You can have multiple lines of above command to have different sets of values for different set of modules.

---

**NOTE**

This command is only applicable for Brocade NetIron MLX Series and Brocade NetIron XMR Series devices.

---

## **Configuring the default action**

By default, the device drops packets that do not match any of the programmed flows. However, you can configure a device-level option to forward the packets to the controller instead of dropping them. This is an optional configuration. If this option is not configured, packets that do not match any flow

entries on a port are dropped. When sending a packet to the controller, a copy of the packet is sent to each of the configured controller connections.

To enable the default action, enter the following command:

```
device(config)# openflow default send-to-controller
```

**Syntax: [no] openflow default send-to-controller**

Packets that match a flow entry on a port are processed according to the action specified and are not affected by this setting. Use the **no** form of the command to set the default action to drop such packets instead.

## Displaying the OpenFlow status on the device

After enabling or disabling OpenFlow on a device, you can verify the configuration using any of the **show** commands.

***Use the show running configuration command:***

When OpenFlow is enabled on the device, the **show running configuration** command displays output similar to the following:

```
device(config)# show run
Current configuration:
ver V5.4.0iT183
mirror ethernet 1/19
openflow enable ofv100
```

***Displaying Openflow show***

The **show OpenFlow** command displays the OpenFlow configuration. It includes the configured unprotected VLANs as well.

```
device(config)# show openflow

Administrative Status:      Enabled
SSL Status:                 Enabled
Controller Type:           OFV 100
Number of Controllers:     1
Controller 1:
Connection Mode:          passive, TCP,
Listening Address:        0.0.0.0
Connection Port:          6633
Connection Status:        TCP_LISTENING
Match Capability:
L2 : Port, Source MAC, Destination MAC, Ether type, Vlan, Vlan PCP
L3 : Port, Vlan, Vlan PCP, Ethertype(IP,ARP,LLDP), Source IP, Destination IP, IP
Protocol, IP TOS, IP Src Port, IP Dst Port
L23: All
Normal Openflow Enabled Ports:
Openflow Hybrid Interfaces:
e1/1
Protected VLANs   : None
Unprotected VLANs :    2, 3, 4, 5, 6, 7, 8, 9, 10, 11
.....
3994, 3995, 3996, 3997, 3998, 3999, 4000, 4001, 4011,
e2/1
Protected VLANs   : None
Unprotected VLANs :  4010,
Default action: drop
Maximum number of flows allowed: 65536
Active flow: 0
Maximum number of Protected Vlan allowed: 2048
```

```
Maximum number of Unprotected Vlans allowed: 4096
Total number of Unprotected Vlans: 4002
```

**Syntax: show openflow**

## Displaying the OpenFlow status

If OpenFlow is enabled on a device, you can get a detailed report of the OpenFlow status on that device.

### ***Brocade NetIron CER Series and Brocade NetIron CES Series devices:***

On these devices, only Layer 2 matching mode is supported on the interfaces.

```
device(config)# show openflow interface
```

```
Port Link Port-State Speed Tag MAC OF-portid Name Mode
1/1 Up Forward 1G No 0000.00b4.89c1 1 Layer2
1/2 Up Forward 1G No 0000.00b4.89c2 2 Layer2
1/3 Up Forward 1G No 0000.00b4.89c3 3 Layer2
```

### ***Brocade NetIron MLX Series and Brocade NetIron XMR Series devices:***

```
device(config)# show openflow interface
```

```
Port Link Port-State Speed Tag MAC OF-portid Name Mode
1/5 Up Forward 1G No 0000.0088.0904 5 OpenFlow-A Layer2
1/7 Up Forward 1G No 0000.0088.0906 7 OpenFlow-B Layer3
1/8 Up Forward 1G No 0000.0088.0907 8 OpenFlow-C Layer3
2/1 Up Forward 1G Yes 0000.0088.0916 49 OpenFlow-E Layer2
```

**TABLE 7** Output fields of the show openflow interface command

Field	Description
Port	Indicates the port number on the device.
Link	Indicates the link status.
Port-State	Indicates the action to be performed on packets that reach the interface. Supported actions include forward the packet, modify the VLAN tag, drop the packet, and send to the controllers.
Speed	Indicates the port speed.
Tag	Indicates whether the interface can accept tagged packets or not.
MAC	Indicates the MAC address of the port.
OF-PortID	Indicates the OpenFlow port ID that is assigned to the port on the device. Port numbers on the device are mapped to OpenFlow port IDs. For more information, see <a href="#">Behavior of ports and devices</a> on page 45.
Name	Indicates the name assigned to the port.
Mode	Indicates the OpenFlow mode enabled on the port. Ports on Brocade NetIron CER Series and Brocade NetIron CES Series devices support only Layer 2 mode. Ports on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices support either Layer 2 or Layer 3 mode.



In addition, you can use the **show interface** command at the interface level. If OpenFlow is enabled on the interface, the OpenFlow status is indicated in the output.

```
device(config-if-e1000-1/1)# show interfaces
GigabitEthernet1/1 is down, line protocol is down
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0000.0034.5060 (bia 0000.0034.5060)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Member of VLAN 1 (untagged), port is in untagged mode, port state is Disabled
  STP configured to ON, Priority is level0, flow control enabled
  OpenFlow enabled, Openflow Index 1, Flow Type Layer2
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
```

## Displaying the configured connections to controllers

Use the **show openflow** command to display the OpenFlow configuration, including the configured connections to controllers on the device.

```
device(config)# show openflow
Administrative Status:      Enabled
Controller Type:           OFV 100
Number of Controllers:     3
Controller 1:
  Connection Mode:         active, TCP
  Controller Address:      10.25.128.243
  Connection Port:         6633
Controller 2:
  Connection Mode:         active, TCP
  Controller Address:      10.25.128.242
  Connection Port:         6633
Controller 3:
  Connection Mode:         passive, TCP
  Listening Address:        0.0.0.0
  Connection Port:         6633
Match Capabilty:
Port, Destination MAC, Vlan, Vlan PCP
Openflow Enabled Ports:   e1/1 e1/2
```

**TABLE 8** Output fields for the show openflow command

Field	Description
Administrative Status	Indicates the administrative status of OpenFlow on the device.
Controller Type	Indicates the OpenFlow protocol version that is supported on the device.
Number of Controllers	Lists the number of controller connections configured on the device. Brocade devices support up to three concurrent controller connections.
Connection mode	Indicates the mode of the controller connection configured. You can configure active or passive connection to controllers. An active connection is initiated by the device. In a passive connection, the device is in the listening mode, and accepts requests from controllers. If the optional controller address is not specified, any controller can establish a connection with the device in the passive mode. Refer to <a href="#">Connecting to an OpenFlow controller</a> on page 36.
Controller address	Indicates the address of the specified controller.
Connection port	Indicates the TCP port that is used for connection to the controller. By default, port 6633 is used.
Match capability	Specifies the matching rules supported.

**TABLE 8** Output fields for the show openflow command (Continued)

Field	Description
OpenFlow enabled ports	Lists the ports on the device that are enabled for OpenFlow.
OpenFlow hybrid mode ports	Indicates the VLAN IDs.
Default Action	Indicates the default action for packets that do not match any configured flows. By default, such packets are dropped. However, you can configure these packets to be sent to the controller by using the <b>openflow default send-to-controller</b> command.
Maximum Number of Flows Allowed	Indicates the maximum number of flows allowed on the device that is configured by using the <b>system-max openflow-flow-entries</b> command.

## Displaying the data path ID of the device

OpenFlow associates a globally unique data path ID to be used by the controller to distinguish OpenFlow devices on a network. To display the data path ID assigned to the device, enter the following command:

```
device(config)# openflow datapath-id
datapath-id# 0000001bedb3d0c0
```

**Syntax:** **openflow datapath-id**

## Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the device and their statistics by using the following command:

```
device(config)# show openflow flows eth 1/2
```

**Syntax:** **show openflow flows [ eth x/y/z ] [ flowid ]**

The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed. You can use the optional **flowid** keyword to display the flows configured in the system and associated with the specified flow ID.

### NOTE

On the Brocade Netron CER Series and Brocade Netron CES Series devices, statistics are available only for up to 2048 flows on a first-come-first-served basis.

```
device(config)# show openflow flows
Total Number of Flows: 2
Total number of data packets sent to controller: 0
Total number of data bytes sent to controller: 0
Flow ID: 1 Priority: 32768 Status: Active
  Rule:
    In Port: e1/2
    In Vlan: Untagged
    Destination Mac: 000.0000.0001
    Destination Mac Mask: FFFF.FFFF.FFFF
  Action: FORWARD
    Out Port: e1/1, Untagged
```

```

Statistics: 0
  Total Pkts: 0
  Total Bytes: 0
Flow ID: 2 Priority: 32768 Status: Active
Rule:
  In Port: e1/2
  In Vlan: Tagged [10]
  Vlan PCP: 4
  Destination Mac: 0000.0000.0001
  Destination Mac Mask: FFFF.FFFF.FFFF
Action: FORWARD
  Out Port: e1/1, Untagged
  Out Port: e1/3, Untagged
Statistics: 0
  Total Pkts: 0
  Total Bytes: 0

```

**TABLE 9** Output fields for the show openflow flows command

Field	Description
Total Number of Flows	The total number of flows on the device.
Total number of data packets sent to controller	The number of packets sent to the controller.
Total number of data bytes sent to controller	The number of bytes sent to the controller.
Flow ID	An identifier for each flow. You can use the flow ID from this output to display flow-specific details.
Priority	The priority of the flow set by the controller when the flow is added, in the range 0 to 65536. If the priority value was not specified, the Brocade device will assign the default value, 32768.
Status	Indicates whether the flow is configured correctly in the device. A correctly configured flow will have its status as active.
Rule	<p>Specifies the matching rule for the flow. In this example, the matching rule is for the flow with Flow ID 1, to forward untagged packets reaching the interface eth 1/2 with the destination MAC address of 0000.0000.0001, to the egress port eth 1/1 as untagged packets.</p> <p>Here, the destination MAC Address Mask of FFFF.FFFF.FFFF indicates that only packets exactly matching the specified destination MAC address will be forwarded.</p>
Statistics	Indicates the counter of packets and bytes.
<p><b>NOTE</b></p> <p>For Brocade NetIron CER Series and Brocade NetIron CES Series devices, only the number of packets will be displayed.</p>	

## Purge-time for OpenFlow

You can configure the maximum time before stale flows are purged from the OpenFlow flow table after a switch-over, fail-over, or operating system upgrade. The no form of this command sets the purge timer time to its default value.

The valid range is from 1 through 600. The default is 240 seconds.

You may not need to change the value of the OpenFlow purge timer for normal circumstances. If you anticipate delay in learning the flows from controller after switch-over, you can configure a larger value for the OpenFlow purge timer.

The following example shows how to set the OpenFlow purge timer:

```
device(config)# openflow purge-time 500
device(config)# no openflow purge-time 350
```

**Syntax: openflow purge-time seconds**

**Syntax: no openflow purge-time seconds**

# Administrating OpenFlow

## Clearing the OpenFlow statistics

You can clear the flow statistics for all flows or, optionally, for a specified flow. Only the counters of packets and bytes (when applicable) are cleared; none of the other flow table entries are affected.

To clear flow counters, enter the following command:

```
device(config)# clear statistics openflow
```

**Syntax: clear statistics openflow [ flow-id ]**

The **flow-id** keyword, if specified, clears only the counters for the specified flow. Use the **show openflow flows** command to obtain flow IDs.

## Deleting the OpenFlow flows

When an individual OpenFlow rule or all flows in the flow table need to be deleted, you can use the **clear openflow** command. Use this command to delete a single OpenFlow rule based on a Flow ID or delete all flows in the flow table.

```
device# clear openflow flowid 6
```

**Syntax: clear openflow flowid flow-id| all**

The **flowid flow-id** parameter deletes a single OpenFlow rule with the given flow ID. The all keyword deletes all flows in the flow table. The command will delete the rule irrespective of the state it is in (ACTIVE, PENDING\_ADD, PENDING\_MODIFY, or PENDING\_DELETE). The same rule can be added again later from the controller if needed.

## Show tech

The **show tech-support openflow** command captures the output of multiple show commands at one time, to be used for diagnostic purposes.

```
device# show tech-support openflow
```

**Syntax: show tech-support openflow**

You can capture the output of the following commands:

- **show openflow datapath-id**
- **show openflow controller**
- **show openflow interface**
- **show openflow flows**
- **show versions**
- **show interfaces**
- **show statistics**
- **show running-config**
- **show logging**
- **show save**

## OpenFlow configuration considerations

After you enable OpenFlow on a device, you can configure, generate, and monitor flows on the ports configured on the device from a controller on OpenFlow-enabled ports. The Brocade device flow table is entirely under the control of the OpenFlow Controller.

- OpenFlow action can duplicate traffic to 16 ports
- The OpenFlow Controller supports Administratively down ( OFPPC\_PORT\_DOWN) through a Port Modification Message.

### Behavior of ports and devices

- Ports that are enabled for OpenFlow cannot take part in any of the normal operations of the device, such as routing and Layer 2 forwarding. However, after OpenFlow is disabled on a port, the port can resume normal operations. This does not require disabling OpenFlow globally on the device.
- OpenFlow defines port numbers sequentially from 1. The `OF-portid` parameter in the output of the **show openflow interface** command is assigned to the ports on the device. On Brocade NetIron MLX Series and Brocade NetIron XMR Series devices, 48 OpenFlow ports are reserved per slot. OpenFlow port numbering starts from slot 1. That is, OpenFlow port 1 is port 1/1 (1/1 = slot 1/port 1), OpenFlow port 2 is port 1/2, and so on. Therefore, slot 1 has OpenFlow ports 1-48, slot 2 has OpenFlow ports 49-96, and so on.

For example, if slot 1 is an 8x10G card and slot 2 is an 8x10G card, then the OpenFlow ports will be: slot 1 (OpenFlow ports 1 to 8); slot 2 (OpenFlow ports 49 to 56). On Brocade NetIron CER Series and Brocade NetIron CES Series devices, ports 1/1 to 1/48 are OpenFlow ports 1 to 48 and ports 2/1 and 2/2 are OpenFlow ports 49 and 50. The OpenFlow protocol offers a capability discovery message for the controller to discover the ports that are OpenFlow-enabled on the router and their capabilities.

- The flow table content is not cleared when the connection to a controller is lost. The device will continue to forward traffic according to the flow entries defined in the flow table even in the absence of a controller connection.
- The flow table entries within the device are cleared when the device is reset.
- On the Brocade NetIron MLX Series and Brocade NetIron XMR Series devices, when the active management module in the device switches over, all controller connections are closed. Configured controller connections are re-established after the device switches over to the standby management module.
- Flow table entries associated with a port are maintained when a port goes down. When the port comes back up, those flow entries are restored on the port. Flow entries are removed only with an explicit command from the controller.
- When OpenFlow is disabled globally on the device using the **no openflow enable** command, the flow table in the device is cleared. However, before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- When a controller tries to add a flow to the device with the same priority, rule, and action as a flow that exists in the flow table, the flow statistics are cleared (the system does not add a new flow). The following table summarizes the behavior for similar flows being successively added.

**TABLE 10** Flow table behavior when flows similar to existing ones are added

Priority	Rule	Action	Device behavior
Same	Same	Same	Clear flow statistics
Same	Same	Different	<ul style="list-style-type: none"> <li>• Update the action list</li> <li>• Clear the statistics</li> </ul>
Same	Different	Same	Create new flow
Same	Different	Different	Create new flow
Different	Same	Same	Create new flow
Different	Same	Different	Create new flow

## Removing an OpenFlow configuration from a device

In general, to remove OpenFlow from the device and make it a non-OpenFlow device, complete the following steps:

1. Disable OpenFlow on the ports where it is enabled.
2. Disable OpenFlow on the device globally.
3. (Optional) Set the maximum number of flows to zero using the **system-max openflow-flow-entries 0** command.
4. Reload the device.

# OpenFlow 1.3

---

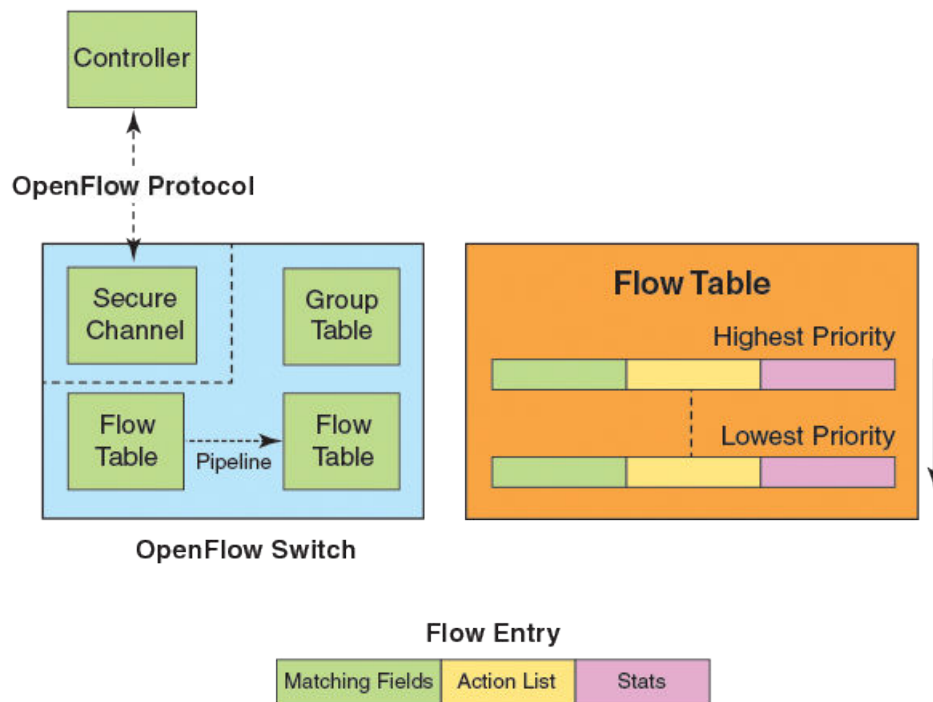
- Overview of OpenFlow 1.3.....47
- Group table..... 72
- QinQ..... 76
- Enqueue..... 82
- Metering..... 85
- TTL support..... 88
- Forwarding port and controller for matching ARP..... 89
- sFlow support on OpenFlow ports..... 89
- TLS1.2 support for NetIron devices..... 90

## Overview of OpenFlow 1.3

An OpenFlow switch maintains one or more flow tables, which are used for packet processing. The switch performs the actions listed in the table entry corresponding to the matched flow.

The OpenFlow Controller manages the OpenFlow switch using the OpenFlow Protocol. The OpenFlow Controller can add, delete, or modify flows by getting statistics for ports and flows and other information using the OpenFlow Protocol.

**FIGURE 5** OpenFlow 1.3 architecture

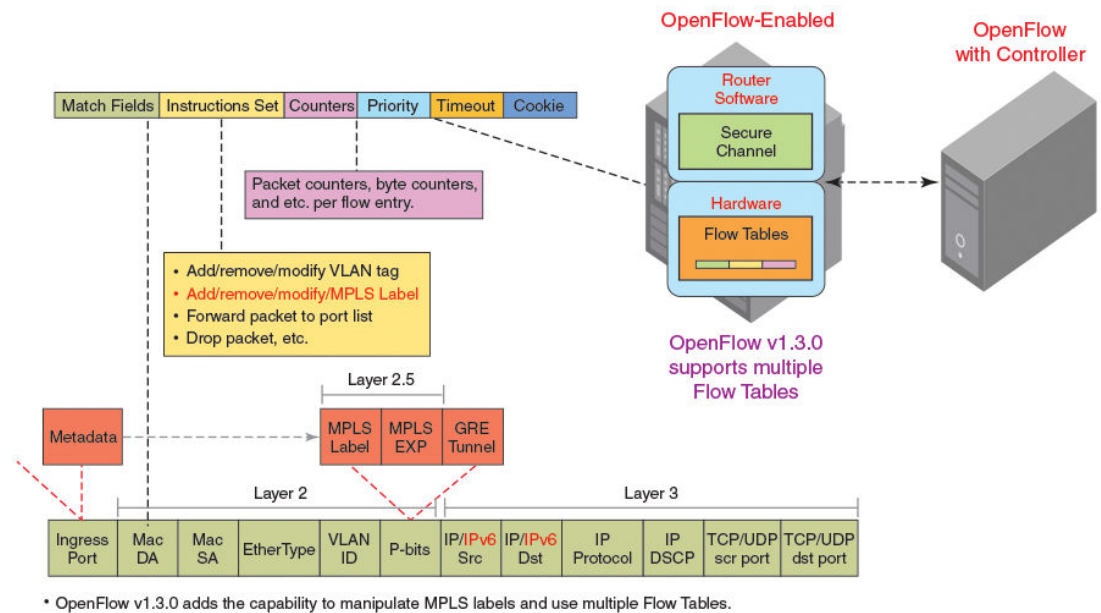


Each flow table maintained in a switch, consists of flow entries sorted by the flow priority. Highest priority flows are at the top of the flow table. Incoming packets are matched against the flow entries starting from the highest priority flow. If there is a match, then flow matching stops, and the set of actions for that flow entry performed. The packets that don't match any flow entry, are either dropped, or sent to the controller.

OpenFlow 1.3 defines three tables:

- Flow tables
- Group table
- Meter table

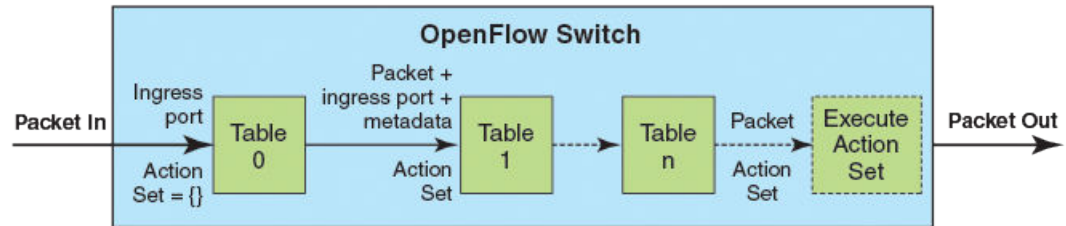
**FIGURE 6** OpenFlow 1.3 flow table entries



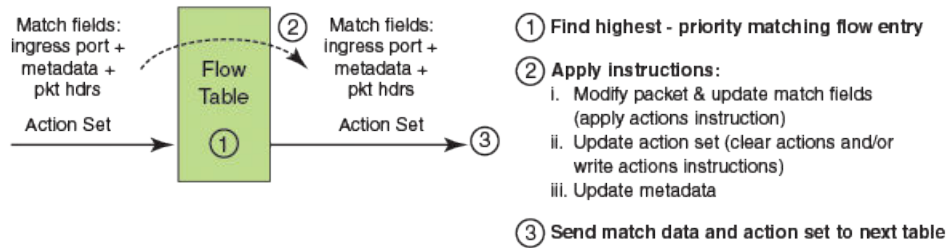
The incoming packets are matched against the multiple tables in the pipeline.



**FIGURE 7** Pipeline processing



(a) Packets are matched against multiple tables in the pipeline



(b) Per-table packet processing

## OpenFlow IPv6 rule support

This feature enables support for matching IPv6 tuples for OpenFlow flows.

The match fields applicable for IPv6 traffic are shown below. OpenFlow supports matching of IPv6 tuples setup for these fields. When a match is found the corresponding action can be executed.

### NOTE

The existing layer 2 match fields in each individual port mode are all supported along with IPv6 match fields on Brocade NetIron MLX Series and Brocade NetIron XMR Series along with these match fields.

### NOTE

IPv6 support on Layer23 port will match only the IPv6 Layer 3 header fields and VLAN tag on Brocade NetIron CER Series and Brocade NetIron CES Series

**TABLE 11** IPv6 OpenFlow match fields

Match field	Brocade NetIron MLX Series	Brocade NetIron CER Series	Pre-requisite	Description
OFPXMT_OFB_IP_DSCP	Yes	Yes	ETH_TYPE=0x0800 or 0x86dd	IPv6 traffic class
OFPXMT_OFB_IP_ECN	No	No	ETH_TYPE=0x0800 or 0x86dd	IPv6 traffic class

**TABLE 11** IPv6 OpenFlow match fields (Continued)

Match field	Brocade NetIron MLX Series	Brocade NetIron CER Series	Pre-requisite	Description
OFPXMT_OFB_IP_PROTO	Yes	Yes	ETH_TYPE=0x0800 or 0x86dd	IPv6 next header
OFPXMT_OFB_TCP_SRC	Yes	Yes	IP PROTO = 6	TCP source port for IPv6 packets
OFPXMT_OFB_TCP_DST	Yes	Yes	IP PROTO = 6	TCP destination port for IPv6 packets
OFPXMT_OFB_UDP_SRC	Yes	Yes	IP PROTO = 17	UDP source port for IPv6 packets
OFPXMT_OFB_UDP_DST	Yes	Yes	IP PROTO = 17	UDP destination port for IPv6 packets
OFPXMT_OFB_IPV6_SRC	Yes	Yes	ETH_TYPE=0x86dd	IPv6 source address
OFPXMT_OFB_IPV6_DST	Yes	Yes	ETH_TYPE=0x86dd	IPv6 destination address
OFPXMT_OFB_IPV6_FLABEL	No	No	ETH_TYPE=0x86dd	IPv6 flow label
OFPXMT_OFB_ICMPV6_TYPE	Yes	Yes	IP PROTO = 58	IPv6 ND type
OFPXMT_OFB_ICMPV6_CODE	Yes	Yes	IP PROTO = 58	IPv6 ND code
OFPXMT_OFB_IPV6_ND_TARGET	No	No	ICMPv6 type=135 or 136	IPv6 ND field
OFPXMT_OFB_IPV6_ND_SLL	No	No	ICMPv6 type=135	IPv6 ND field
OFPXMT_OFB_IPV6_ND_TLL	No	No	ICMPv6 type=136	IPv6 ND field
OFPXMT_OFB_IPV6_EXTHDR	No	No	IP PROTO = 58	IPv6 extended header pseudo-field
OFPXMT_OFB_ICMPV4_TYPE	Yes	Yes	IP PROTO = 1	ICMPv4 type
OFPXMT_OFB_ICMPV4_CODE	Yes	Yes	IP PROTO = 1	ICMPv4 code

### ***Actions***

All flow actions for IPv4 matching flows are supported for IPv6 matching flows as well.

### ***Match fields supported on Brocade NetIron CER & Brocade NetIron CES series***

The following table shows the supported rules on different types of OpenFlow ports.

**TABLE 12** Match fields supported

Match field	Layer23	Layer 3
Source port	Yes	Yes

**TABLE 12** Match fields supported (Continued)

Match field	Layer23	Layer 3
MAC Destination	Yes <sup>1</sup>	No
MAC Source	Yes <sup>1</sup>	No
VLAN ID	Yes	Yes
VLAN PCP	Yes	Yes
ETH_TYPE	Yes	Yes
IPv4 SIP	Yes	Yes
IPv4 DIP	Yes	Yes
IPv6 SIP	Yes	Yes
IPv6 DIP	Yes	Yes
IP Protocol	Yes	Yes
L4 Source port	Yes	Yes
L4 Destination port	Yes	Yes
IP DSCP	Yes	Yes
IPv6 Traffic class	Yes	Yes
ICMPv6 Type	Yes	Yes
ICMPv6 Code	Yes	Yes
ICMPv4 Type	Yes	Yes
ICMPv4 Code	Yes	Yes

**NOTE*****Limitations***

- Ethertype is compulsory in the OpenFlow rules for Layer 3 and Layer23 ports for Brocade NetIron CER Series and Brocade NetIron CES Series.
- IPv6 matching field is not supported in this release for Brocade NetIron CER Series and Brocade NetIron CES Series. Any flow addition or modification with IPv6 matching field will be rejected when the input port is configured in OpenFlow Layer 2 mode.

***IPv6 packets***

IPv6 packets consist of IPv6 header and upper layer protocol data unit (PDU) and sometimes extension headers as well. The size of the IPv6 header is 40 bytes. The fields in the IPv6 header are: Source IP, Destination IP, Next Header (IP Protocol), Traffic class, Flow label, Hop Limit. A Next Header field in the

<sup>1</sup> 1: Except for IPv6 Ethertype

IPv6 header indicates the next extension header. IPv6 extension headers have no maximum size and can expand to accommodate all the extension data needed for IPv6 communication.

The upper layer protocol data unit (PDU) usually consists of an upper layer protocol header and its payload, for example an ICMPv6 message, a UDP message, or a TCP segment.

This table shows the typical value of the Next Header field for an IPv6 header or an IPv6 extension header.

**TABLE 13** Next Header fields

Value (in decimal)	Header
0	Hop-by-Hop Options Header
6	TCP
17	UDP
41	Encapsulated IPv6 Header
43	Routing Header
44	Fragment Header
46	Resource Reservation Protocol
50	Encapsulating Security Payload
51	Authentication Header
58	ICMPv6
59	No next header
60	Destination Options Header

### ***Scalability for IPv6***

To support IPv6 matching, hardware entries are reserved in IPv6 session and IPv6 SuperACL partitions.

System-max command **np-openflow-entries** is modified to support IPv6.

### **Scaling for Brocade NetIron MLX Series and Brocade NetIron XMR Series**

The following parameters are available for this command:

- layer3IPv6-To add IPv6 matching flows on OpenFlow layer 3 ports
- layer23IPv6-To add IPv6 matching (along with Layer 2 fields) flows on OpenFlow layer23 ports

---

#### **NOTE**

IPv6 SuperACL is not supported on Gen1 cards and Interface modules. So layer23IPv6 command will be rejected for these devices.

---



---

#### **NOTE**

layer2or3, layer23IPv4, layer3IPv6 and layer23IPv6 configurations are independent of each other and should be in separate lines for configuration generations.

---

**TABLE 14** System-max matching fields

System-max	Matching fields
layer2or3	Layer 2 or Layer 3 fields
layer23IPv4	Layer 2 and Layer 3 fields (non IPv6 traffic)
layer3IPv6	Layer 3 fields for IPv6 traffic
layer23IPv6	Layer 2 and Layer 3 fields for IPv6 traffic

```
system-max np-openflow-entries {layer2or3 | layer23ipv4 | layer3ipv6 | layer23ipv6}
slot [i j k | <i to z> | <all>]
```

### Scaling for Brocade Netron CER Series and Brocade Netron CES Series

**TABLE 15** System-max matching fields

Device	Layer 2	Layer23	Layer 3
CES series <sup>1</sup>	4k	2k	2k
CER series <sup>2</sup>	32k	2k	2k

### *OpenFlow interface support on hybrid and non-hybrid ports*

These are the rules supported on Brocade Netron MLX Series and Brocade Netron XMR Series for both hybrid and non-hybrid OpenFlow ports.

#### **For OpenFlow Layer 2 ports:**

- OpenFlow Layer 2 interfaces can not have flows matching IPv6 fields.

#### **For OpenFlow Layer 3 ports:**

- OpenFlow Layer 3 interfaces can have flows matching IPv6 fields similar to IPv4 fields. OpenFlow Layer 3 interface expects ethertype in flow matching. So, a flow can match one of IPv4, IPv6 or non-IP (LLDP, ARP).

#### **For OpenFlow Layer23 ports:**

- OpenFlow Layer23 interfaces can have flows matching IPv6 fields similar to IPv4 fields. OpenFlow Layer23 interface can have flows matching Layer 2 fields along with IPv6 fields.

#### **NOTE**

Flow matching of IPv6 fields is not supported on Layer23 ports on Brocade Netron CER Series and Brocade Netron CES Series.

#### **NOTE**

Brocade Netron CER Series and Brocade Netron CES Series do not support hybrid port mode on OpenFlow ports.

<sup>2</sup> 1: Resources are shared between Layer 2, Layer3 and Layer23.

<sup>3</sup> 2: Resources are shared between Layer 3 and Layer23.

### ***CAM partition for Brocade NetIron MLX Series and Brocade NetIron XMR Series***

This section indicates maximum available IPv6 Session and IPv6 SuperACL CAM space for OpenFlow.

Total available CAM space is divided into following parts.

- OpenFlow flows
- OpenFlow protected VLANs
- OpenFlow unprotected VLANs (only for Interface modules)

These are the rules followed for the CAM partition for OpenFlow.

1. IPv6 rule ACL need 128 entries by default.
2. CAM sub-partition is not needed for unprotected VLANs except for Interface modules.
3. **ipv6-mcast-cam** system-max value is set to 0 to maximize IPv6 Session and IPv6 SuperACL CAM usage. Default value for this system-max is 2048.
4. IPv6 SuperACL use CAM space from IPv6 Session. Default value for this system-max is 0.

---

#### **NOTE**

For detailed information on CAM partition for IPv6 Session and IPv6 SuperACL, please refer Brocade NetIron Administration Guide.

---

### ***CAM partition for OpenFlow***

CAM is partitioned on the device by a variety of profiles that you can select depending on your application. The available profiles for Brocade NetIron XMR Series and Brocade NetIron MLX Series are listed in Brocade NetIron Administration Guide.

To implement a CAM partition profile, enter the following command.

```
device(config) cam-partition profile ipv4
```

**Syntax:** **cam-partition profile** [ **ipv4** | **ipv4-ipv6** | **ipv4-ipv6-2** | **ipv4-vpls** | **ipv4-vpn** | **ipv6** | **I2-metro** | **I2-metro-2** | **mpls-l3vpn** | **mpls-l3vpn-2** | **mpls-vpls** | **mpls-vpls-2** | **mpls-vpn-vpls** | **multi-service** | **multi-service-2** | **multi-service-3** | **multi-service-4** ]

The *ipv4* parameter adjusts the CAM partitions, for Brocade NetIron XMR Series and for Brocade NetIron MLX Series, to optimize the device for IPv4 applications.

The *ipv4-ipv6* parameter adjusts the CAM partitions, for Brocade NetIron XMR Series and for Brocade NetIron MLX Series, to optimize the device for IPv4 and IPv6 dual stack applications.

The Openflow CAM partition profiles for IPv6 ACL session for Generation 1 and Generation 2 modules are displayed below.

**TABLE 16** Generation 1 CAM partitioning profiles available for Brocade MLX Series routers, Brocade NetIron XMR Series routers, and BR-MLX-10GX24-DM modules

Profile	IPv6 ACL	IPv6 ACL	IPv6 ACL
	Brocade MLX Series routers	Brocade NetIron XMR Series routers	BR-MLX-10GX24-DM modules
			<b>NOTE</b> 180 bits per CAM are supported. Two CAMs equal 1 IPv6 session CAM.
Default Profile	2K	4K	16K
ipv4 Profile	0	0	0
ipv6 Profile	12K	24K	32K
l2-metro Profile	0	0	0
l2-metro-2 Profile	0	0	0
multi-service Profile	4K	8K	16K
multi-service-2 Profile	2K	4K	8K
multi-service-3 Profile	4K	8K	16K
multi-service-4 Profile	4K	8K	16K
multi-service-5 Profile	4K	8K	8K
mpls-vpn-vpls Profile	0	0	0
mpls-l3vpn Profile	0	0	0
mpls-l3vpn-2 Profile	0	0	0
mpls-vpls Profile	0	0	0
mpls-vpls-2 Profile	0	0	0
ipv4-vpn Profile	0	0	0
ipv4-ipv6 Profile	10K	20K	8K
ipv4-vpls Profile	0	0	0
ipv4-ipv6-2Profile	4K	8K	16K
Entry size (in bits)	640	640	360

**TABLE 17** Generation 2 CAM partitioning profiles for IPv6 ACL are displayed in the columns below for each module

Profile	IPv6 ACL BR- MLX-10G X8-M	IPv6 ACL BR- MLX-10GX8 -X	IPv6 ACL NI- MLX-10GX8 -D	IPv6 ACL/ Multicast IPv6 ACL BR- MLX-100GX1- X(4) BR- MLX-100GX2- X(4)	IPv6 ACL BR- MLX-40GX 4-M	IPv6 ACL BR- MLX-10GX20- M (1G/10G combo), BR- MLX-1GX20- U10G-M and BR- MLX-100GX2- CFP2-M	IPv6 ACL BR- MLX-10GX20 -X2 (1G/10G combo), BR- MLX-1GX20- U10G-X2 and BR- MLX-100GX2 -CFP2-X2
Default Profile	2K	4K	2K	16K	6K	4K	12K
ipv4 Profile	0	0	0	0	0	0	4K
ipv6 Profile	12K	24K	12K	48K	20K	24K	60K
I2-metro Profile	0	0	0	0	0	0	0
I2-metro-2 Profile	0	0	0	0	0	0	0
multi-service Profile	4K	8K	4K	32K	6K	8K	20K
multi-service-2 Profile	2K	4K	2K	16K	8K	4K	12K
multi-service-3 Profile	4K	8K	4K	24K	8K	8K	20K
multi-service-4 Profile	4K	8K	4K	32K	8K	8K	24K
multi-service-5 Profile	4K	8K	8K	16K	4K	8K	28K
multi-service-6 Profile	12K	12K	4K	32K	8K	8K	24K
telemetry-1 Profile	12K	12K	6K	24K	12K	20K	28K



**TABLE 17** Generation 2 CAM partitioning profiles for IPv6 ACL are displayed in the columns below for each module (Continued)

Profile	IPv6 ACL BR- MLX-10G X8-M	IPv6 ACL BR- MLX-10GX8 -X	IPv6 ACL NI- MLX-10GX8 -D	IPv6 ACL/ Multicast IPv6 ACL BR- MLX-100GX1- X(4) BR- MLX-100GX2- X(4)	IPv6 ACL BR- MLX-40GX 4-M	IPv6 ACL BR- MLX-10GX20- M (1G/10G combo), BR- MLX-1GX20- U10G-M and BR- MLX-100GX2- CFP2-M	IPv6 ACL BR- MLX-10GX20 -X2 (1G/10G combo), BR- MLX-1GX20- U10G-X2 and BR- MLX-100GX2 -CFP2-X2
mpls- vpn-vpls Profile	0	0	0	0	0	0	0
mpls- l3vpn Profile	0	0	0	0	0	0	0
mpls- l3vpn-2 Profile	0	0	0	0	0	0	0
mpls- vpls Profile	0	0	0	0	0	0	0
mpls- vpls-2 Profile	0	0	0	0	0	0	0
ipv4-vpn Profile	0	0	0	0	0	0	0
ipv4- ipv6 Profile	10K	20K	10K	40K	18K	20K	36
ipv4-vpls Profile	0	0	0	0	0	0	0
ipv4- ipv6-2Pr ofile	4K	8K	4K	16K	8K	8K	4
Entry size (in bits)	640	640	640	320	640	640	640

**TABLE 18** CAM partitioning profiles for IPv6 SuperACL (shared with IPv6 ACL) are displayed in the columns below for each module

Profile	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX8-M	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX8-X	IPv6 SuperACL (shared with IPv6 ACL) NI-MLX-10GX8-D	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-40GX4-M	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
Default Profile	0K	0 - 2K	0K	0 - 6K	0- 4K	0- 2K
ipv4 Profile	0	0	0	0	0	0
ipv6 Profile	0-10K	0-22K	0-10K	0-22K	0 - 18K	0-22K
I2-metro Profile	0	0	0	0	0	0
I2-metro-2 Profile	0	0	0	0	0	0
multi-service Profile	0 - 2K	0 - 6K	0 - 2K	0 - 14K	0 - 4K	0 - 6K
multi-service-2 Profile	0	0 - 2K	0	0 - 6K	0 - 6K	0 - 2K
multi-service-3 Profile	0 - 2K	0 - 6K	0 - 2K	0 - 10K	0 - 6K	0 - 6K
multi-service-4 Profile	0 - 2K	0 - 6K	0 - 2K	0 - 14K	0 - 6K	0 - 6K

**TABLE 18** CAM partitioning profiles for IPv6 SuperACL (shared with IPv6 ACL) are displayed in the columns below for each module (Continued)

Profile	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX8-M	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX8-X	IPv6 SuperACL (shared with IPv6 ACL) NI-MLX-10GX8-D	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-40GX4-M	IPv6 SuperACL (shared with IPv6 ACL) BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
multi-service-5 Profile	0 - 2K	0 - 6K	0 - 6K	0 -6K	0 -2K	0 - 6K
mpls-vpn-vpls Profile	0	0	0	0	0	0
mpls-l3vpn Profile	0	0	0	0	0	0
mpls-l3vpn-2 Profile	0	0	0	0	0	0
mpls-vpls Profile	0	0	0	0	0	0
mpls-vpls-2 Profile	0	0	0	0	0	0
ipv4-vpn Profile	0	0	0	0	0	0
ipv4-ipv6 Profile	0 - 8K	0 -18K	0 -8K	0 -18K	0 -16K	0 -18K
ipv4-vpls Profile	0	0	0	0	0	0

**TABLE 18** CAM partitioning profiles for IPv6 SuperACL (shared with IPv6 ACL) are displayed in the columns below for each module (Continued)

Profile	IPv6 SuperACL (shared with IPv6 ACL)	IPv6 SuperACL (shared with IPv6 ACL)	IPv6 SuperACL (shared with IPv6 ACL)	IPv6 SuperACL (shared with IPv6 ACL)	IPv6 SuperACL (shared with IPv6 ACL)	IPv6 SuperACL (shared with IPv6 ACL)
	BR-MLX-10GX8-M	BR-MLX-10GX8-X	NI-MLX-10GX8-D	BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	BR-MLX-40GX4-M	BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M  BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
ipv4-ipv6-2Profile	0 -2K	0 -6K	0 -2K	0 -6K	0 -6K	0 -6K
Entry size (in bits)	640	640	640	640	640	640

### Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the devices with IPv6 fields. Enter the following command:

The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed. Below is an example to show the IPv6 fields for both in MP and LP.

```
device(config)# show openflow flows
Flow ID: 8 Priority: 32768 Status: Active
Rule:
  In Port:      generic
  Ether type:  0x000086dd
  IP Protocol:  58
  Source IPv6:  2001:db8:85a3:42:0:8a2e:370:7334/128
  Destination IPv6:  2001:2:6c::430/48
  ICMPV6 Type:  1
  ICMPV6 Code:  3
Instructions: Apply-Actions
  Action: FORWARD
  Out Port:    e1/15
Statistics:
  Total Bytes: 0
```

**show openflow flows** have ICMPv4 fields in match for both in MP and LP as following.

```
device(config)# show openflow flows
Flow ID: 6 Priority: 32768 Status: Active
```

```

Rule:
  In Port:      generic
  Ether type:   0x00000800
  Source IP:    51.1.1.1      Subnet IP:    255.255.255.255
  Destination IP: 40.1.1.1      Subnet IP:    255.255.255.0
  IP Protocol:  1
  ICMPV4 Type:  3
  ICMPV4 Code:  2
Instructions: Apply-Actions
  Action: FORWARD
  Out Port:    e1/16
Statistics:
  Total Bytes: 0

```

The **show openflow flows** command has the match capability as below.

Layer 2: Port, Source MAC, Destination MAC, Ether type, VLAN, VLAN PCP

Layer 3 : Port, VLAN, VLAN PCP, Ethertype (IP, IPv6 ,ARP and LLDP), Source IP, Destination IP, IP Protocol, IP TOS , IP Source Port, IP Destination Port, Source IPv6, Destination IPv6, ICMPv6 Type, ICMPv6 code ICMPv4 Type, ICMPv4 code

Layer23: All

---

#### NOTE

Depending on the matching rules involving IPv4 or IPv6 fields, IP Protocol represents both IPv4 and IPv6 protocol numbers.

---



---

#### NOTE

Matching IP TOS supports both IPv4 DSCP and IPv6 traffic class.

---

## Flow table entries

Each flow table entry contains the fields described in the following table.

**TABLE 19** Flow table entries

Field	Description
Match fields	The match fields consist of ingress ports, packet header fields, and metadata from a previous flow table
Priority	Matching precedence of the entry
Counters	Statistics for matching packets
Instructions	Action set or pipeline processing
Cookie	Opaque data sent by the OpenFlow Controller

The following match fields are supported.

- All Layer 2 header fields
- All Layer 3 header fields

**TABLE 20** OpenFlow match fields

Match field	Brocade Netron MLX Series	Brocade Netron CER Series	Pre-requisite	Description
OXM_OF_IN_PORT	Yes	Yes	None	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_ETH_DST	Yes	Yes	None	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	Yes	None	Ethernet source MAC address
OXM_OF_ETH_TYPE	Yes	Yes	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes	None	VLAN-ID 802.1Q header
OXM_OF_VLAN_PCP	Yes	Yes	None	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	Yes	Yes	ETH_TYPE=0x0800	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	Yes IPv4 only	Yes IPv4 only	ETH_TYPE=0x0800	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	Yes	Yes	ETH_TYPE=0x0800	IPv4 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV4_DST	Yes	Yes	ETH_TYPE=0x0800	IPv4 destination address. It can use subnet mask or arbitrary bit mask.
OXM_OF_TCP_SRC	Yes	Yes	IP PROTO = 6	TCP source port
OXM_OF_TCP_DST	Yes	Yes	IP PROTO = 6	TCP destination port
OXM_OF_UDP_SRC	Yes	Yes	IP PROTO = 17	UDP source port
OXM_OF_UDP_DST	Yes	Yes	IP PROTO = 17	UDP destination port

## OpenFlow instructions

Each flow entry has a set of instructions that are executed when the packet matches the entry.

The instruction set associated with each flow entry can have a maximum of one instruction of each type. Following table shows the actions supported on different Brocade devices.

**TABLE 21** Actions for flow table instruction

Actions	Description	Support
Write-Action actions ( Req )	Adds or overwrites specified actions to the action set.	Yes
Apply-Actions actions	Applies the specified actions immediately.	Yes
Clear-Actions actions	Clears all the actions in the action set.	Yes

**TABLE 21** Actions for flow table instruction (Continued)

Actions	Description	Support
Meter meter-id	Directs the packet to the specified meter.	Yes
Goto -Table next-table-id ( Req )	Indicates the next table in pipeline processing.	No <sup>1</sup>
Write-Metadata metadata/ mask	Writes the metadata field from the mask.	No <sup>1</sup>
Output (Req)	Forwards the packet to a specified OpenFlow port. If out-port is Controller, then the packet will be sent as packet-in message.	Yes
Drop (Req)	No explicit drop action. Packet with empty action set should be dropped.	Yes
Group	Processes the packet through the specified group.	Yes
Set field	Modifies the values of the packet header based on the field type.	Yes
Push-Tag/ Pop-Tag	Adds and removes tag (newly inserted tags are always the outermost tags).	Yes
Set-Queue	Enqueues the packet to a specific queue on the outgoing port.	Yes
Change TTL	Modifies the TTL value.	Yes <sup>1</sup>

**NOTE**

1: This action behaves differently on different Brocade devices.

The set fields in the following table are supported for OpenFlow instructions. The set field action is used to set the value in the header field.

**TABLE 22** Supported set field action

Set field	Brocade Netron MLX Series	Brocade Netron CER Series	Brocade Netron XMR Series	Brocade Netron CES Series	Description
OXM_OF_ETH_DST	Yes	No			Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	No			Ethernet source MAC address
OXM_OF_ETH_TYPE	No	No			Ethernet type of the OpenFlow packet payload after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes			VLAN-ID 802.1Q header
OXM_OF_VLAN_PCP	Yes	Yes			VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	Yes	Yes			Diff Serv Code Point (DSCP). Part of the IPv4 ToS field or the IPv6 Traffic Class field.

## OpenFlow actions

Each flow has a set of instructions that are executed when the packet matches the flow as per OpenFlow 1.3 specifications. Each flow can have a maximum of one instruction of each type.

A switch can reject a flow entry, if it is unable to execute the instructions associated with the flow entry. In this case, the switch returns an unsupported flow error. Flow tables may not support every match, every instruction, or every action.

**TABLE 23** Instructions for OpenFlow actions

Instruction	Description
actions	Adds specified actions to the action set
next-table-id	Indicates the next table in pipeline processing (One table is supported.)
meter-id	Directs the packet to the specified meter
apply-actions	Applies the specified actions immediately. The packet is modified and subsequent matching in the pipeline is done on the modified packet.
clear-actions	Clears all the actions in the action set
write-metadata	Writes the metadata field from the mask

Brocade devices may support the actions listed in the following table.

**TABLE 24** OpenFlow actions supported on Brocade devices

OpenFlow action	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Process the packet through the specified group	Yes	Yes
Add and remove tag	Yes	Yes
Add newly inserted tags always as the outermost tags	Yes	Yes
No explicit drop action. Packet with empty action set should be dropped.	Yes	Yes
Modify the values of the packet header based on the field type	Yes	Yes
Modify the TTL value	Yes	No
Set the queue ID for the packet	Yes	No

## Multiple controller connections

An OpenFlow switch may be connected to multiple controllers for reliability. It allows the switch to continue to operate in OpenFlow mode if a controller or controller connection fails. The controllers coordinate the management of the switch amongst themselves to help synchronize controller handoffs.

Each controller can have one of the following roles:



- Equal - The controller has full access to the switch. It can receive all the asynchronous messages from the switch and send commands to modify the state of the switch (add or delete flows).
- Slave - The controller has a read-only access to the switch. It does not receive the asynchronous messages (apart from port status). It is denied the ability to execute commands that modify the state of the switch: **packet-out**, **flow-mod**, **group-mod**, **port-mod**, or **table-mod**. The switch must reply with an OFPT\_ERROR message, if it receives one of those commands from a Slave controller. Other controller-to-switch messages are processed normally.
- Master - The controller has full access to the switch as in the Equal role. When the controller changes its role to Master, the switch changes the other controller in the Master role to have the Slave role. The role change does not affect controllers with the Equal role.

A switch can be simultaneously connected to multiple controllers in the Equal role, multiple controllers in the Slave role, and, at most, one controller in Master role. Each controller can communicate its role to the switch by way of an OFPT\_ROLE\_REQUEST message. This message can be used by the controller to set and query the role of its channel with the switch.

To detect the out-of-order messages during a master-to-slave transition, the OFPT\_ROLE\_REQUEST message contains a 64-bit generation ID, filed by sequence number, that identifies the mastership view. The controllers coordinate the assignment of generation IDs. The generation ID is a monotonically increasing counter. A new (larger) value is assigned each time the mastership view changes; that is, when a new master is designated. The generation ID value wraps around once the maximum value has been reached.

```
device(config)# openflow controller
```

Contlr	Mode	TCP/SSL	IP-address	Port	Status	Role
1	(Equal)	passive	TCP	0.0.0.0	6633	TCP_LISTENING
2	(Master)	active	TCP	10.25.128.179	6633	OPENFLOW_ESABLISHED
3	(Slave)	active	TCP	10.25.128.177	6633	OPENFLOW_ESABLISHED
3	(Equal)	active	TCP	10.25.128.165	6633	OPENFLOW_ESABLISHED

## Asynchronous configuration

Asynchronous messages may need to be sent to multiple controllers. An asynchronous message is duplicated for each eligible OpenFlow channel, and each message is sent when the respective controller connection allows it.

A controller can also control which types of switch asynchronous messages are sent over its OpenFlow channel. This is done using an asynchronous configuration message that has the filter setting for all the messages.

Different controllers can receive different notifications. A controller in the Master role can selectively disable notifications, and a controller in the Slave role can enable notifications it wants to monitor.

Each controller configuration block for active connection maintains its own asynchronous configuration setting for every role. The default initial configuration is shown in the following table.

**TABLE 25** Action for asynchronous configuration

Messages	Bit field	Master or Equal role	Slave role
Packet-in reasons	NO_MATCH	Enable	Disable
	ACTION	Enable	Disable
	INVALID_TTL	Enable	Disable

**TABLE 25** Action for asynchronous configuration (Continued)

Messages	Bit field	Master or Equal role	Slave role
Port status reasons	ADD	Enable	Enable
	DELETE	Enable	Enable
	MODIFY	Enable	Enable
Flow removed reasons	IDLE_TIMEOUT	Enable	Disable
	HARD_TIMEOUT	Enable	Disable
	DELETE	Enable	Disable
	GROUP_DELETE	Enable	Disable

**NOTE**

The asynchronous messages ACTION, INVALID\_TTL, IDLE\_TIMEOUT, and HARD\_TIMEOUT are not supported by Brocade devices. Controllers can set these bits in the filter setting and the device can accept the bits, but the messages will not be sent out by the device.

## Configuring source-interface for the controllers

This feature allows you to configure a source-interface to be used by the device for the connection to the controllers.

Currently, Brocade devices support up to 3 active connections and 1 passive connection to the controllers. Controllers can be configured using the CLI interface of the routers. On configuring an active connection to a controller, the device tries and establishes a connection to the controller. However syslog message will be generated. The connection will be re-attempted every 15 seconds. If the given source-interface has multiple IP addresses configured, then the controller connection will be initiated with the lowest IP address. However, the syslog message is generated only for the first attempt to indicate the error.

- If the source-interface is down:  
The openflow source-interface used for the controller %I is down. Will retry the connection later.
- If there is no IP address configured:  
No IP address configured on the source-interface used for the controller %I. Will retry the connection later.
- When source interface is configured or unconfigured for a controller without force-reconnect option:  
Reset existing active connections to controller(s) for the new Source Interface configuration to take effect.

This feature is also applicable to active connections made using SSL. This feature is not supported on IPv6 interfaces.

To configure a source-interface for the connection from the device to the controller, enter the following command.

```
device(Config)#openflow controller source-interface ethernet 2/2
```

## Supported OpenFlow messages

The following OpenFlow messages are supported on the Brocade devices.

**TABLE 26** OpenFlow messages

Message type	Netron XMR and Netron MLX series	Netron CER and Netron CES series
OFPT_HELLO	Yes	Yes
OFPT_ERROR	Yes	Yes
OFPT_ECHO_REQUEST	Yes	Yes
OFPT_ECHO_REPLY	Yes	Yes
OFPT_EXPERIMENTER	No	No
OFPT_FEATURES_REQUEST	Yes	Yes
OFPT_FEATURES_REPLY	Yes	Yes
OFPT_GET_CONFIG_REQUEST	No	No
OFPT_GET_CONFIG_REPLY	No	No
OFPT_SET_CONFIG	No	No
OFPT_PACKET_IN	Yes	Yes
OFPT_FLOW_REMOVED	Yes	Yes
OFPT_PORT_STATUS	Yes	Yes
OFPT_PACKET_OUT	Yes	Yes
OFPT_FLOW_MOD	Yes	Yes
OFPT_GROUP_MOD	Yes	Yes
OFPT_PORT_MOD	No	No
OFPT_TABLE_MOD	No	No
OFPT_MULTIPART_REQUEST	Yes	Yes
OFPT_MULTIPART_REPLY	Yes	Yes
OFPT_BARRIER_REQUEST	Yes	Yes
OFPT_BARRIER_REPLY	Yes	Yes
OFPT_QUEUE_GET_CONFIG_REQUEST	Yes	No
OFPT_QUEUE_GET_CONFIG_REPLY	Yes	No
OFPT_ROLL_REQUEST	Yes	Yes
OFPT_ROLL_REPLY	Yes	Yes
OFPT_GET_ASYNC_REQUEST	Yes	Yes
OFPT_GET_ASYNC_REPLY	Yes	Yes
OFPT_SET_ASYNC	Yes	Yes
OFPT_METER_MOD	Yes	Yes

## Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the devices with IPv6 fields. Enter the following command:

The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed. Below is an example to show the IPv6 fields for both in MP and LP.

```
device(config)# show openflow flows
Flow ID: 8 Priority: 32768 Status: Active
Rule:
  In Port:      generic
  Ether type:  0x000086dd
  IP Protocol: 58
  Source IPv6: 2001:db8:85a3:42:0:8a2e:370:7334/128
  Destination IPv6: 2001:2:6c::430/48
  ICMPV6 Type: 1
  ICMPV6 Code: 3
Instructions: Apply-Actions
  Action: FORWARD
  Out Port: e1/15
Statistics:
  Total Bytes: 0
```

**show openflow flows** have ICMPv4 fields in match for both in MP and LP as following.

```
device(config)# show openflow flows
Flow ID: 6 Priority: 32768 Status: Active
Rule:
  In Port:      generic
  Ether type:  0x00000800
  Source IP:    51.1.1.1      Subnet IP:    255.255.255.255
  Destination IP: 40.1.1.1      Subnet IP:    255.255.255.0
  IP Protocol:  1
  ICMPV4 Type:  3
  ICMPV4 Code:  2
Instructions: Apply-Actions
  Action: FORWARD
  Out Port: e1/16
Statistics:
  Total Bytes: 0
```

The **show openflow flows** command has the match capability as below.

Layer 2: Port, Source MAC, Destination MAC, Ether type, VLAN, VLAN PCP

Layer 3 : Port, VLAN, VLAN PCP, Ethertype (IP, IPv6 ,ARP and LLDP), Source IP, Destination IP, IP Protocol, IP TOS , IP Source Port, IP Destination Port, Source IPv6, Destination IPv6, ICMPv6 Type, ICMPv6 code ICMPv4 Type, ICMPv4 code

Layer23: All

---

### NOTE

Depending on the matching rules involving IPv4 or IPv6 fields, IP Protocol represents both IPv4 and IPv6 protocol numbers.

---



---

### NOTE

Matching IP TOS supports both IPv4 DSCP and IPv6 traffic class.

---

## Normal action

Normal action represents the traditional non-OpenFlow pipeline of the device. It can be used only as an output port and it processes the packet using the normal flows.

OpenFlow flows can be configured with action as normal matching configured unprotected VLAN on an OpenFlow hybrid port. Traffic matching the configured unprotected VLAN on that hybrid port will be subjected to forwarding or routing as per the forwarding table however QoS parameter such as meters will be taken from matching flow entry and applied on the traffic.

---

### NOTE

A flow with Normal action matching un-configured unprotected VLAN or protected VLAN flow will not be accepted.

---

When the device loses contact with all controllers, the device immediately enters into the fail secure mode or fail standalone mode. In fail secure mode, the packets and messages for the controller are dropped for the device. In fail standalone mode, the device processes all packets using the OFPP\_NORMAL reserved port. NetIron devices support only fail secure mode.

### *Configuring a flow with Normal action*

Configuration should be done in the following order for adding flows with Normal Action.

1. Add configured unprotected VLAN on an OpenFlow hybrid port.
2. Add a Flow matching configured unprotected VLAN with action as normal.

### *Considerations*

The following assumptions are taken into consideration for normal action flows.

1. If route or forwarding entry is not present for configured unprotected VLAN, the packet will be dropped even though a matching normal action flow exists.
2. Multiple normal action flows can be configured for the same match VLAN with different QoS action. Traffic will match the highest priority flow among normal action flows.
3. Flow with different action for the same match VLAN as normal action flow can also be configured.
4. When multiple flows are present, traffic will hit the flow with the highest priority.

### *Limitations*

These are the limitation for normal action flows.

1. Normal action is supported only on flow matching configured unprotected VLAN.
2. Output port (physical port, drop or controller) in the action cannot be specified along with normal action.
3. Generic flow with normal action is not supported. Only port based flows are supported.
4. Wild card VLAN match (port based or generic) flows with normal action is not supported.
5. Normal action flow is supported only on OpenFlow hybrid port.
6. Only meter action is supported for normal action flows.

---

### NOTE

For a flow with action other than normal but matching normal action flow in VLAN on an OpenFlow hybrid port, if that flow does not specify Destination MAC, VLAN or VLAN PCP in the action list, these fields will still get updated if there is a matching IP route entry.

---

### Updating show OpenFlow flows output

show openflow flows shows the Normal action in the output.

```
device#show openflow flows
.....
.....
Flow ID: 1 Priority: 32768 Status: Active
  Rule:
    In Port:      e4/1
    In Vlan:      Tagged[23]
  Instructions: Apply-Actions
    Action: FORWARD
      Out Port: NORMAL
  Statistics:
    Total Pkts: 0
    Total Bytes: 0
```

## Supporting untagged VLAN on protected and configured unprotected VLAN

Brocade devices support untagged traffic on protected VLAN or Configured unprotected VLAN supporting IP traffic on OpenFlow hybrid port. You will be able to configure untagged VLAN as protected VLAN or unprotected VLAN.

This feature supports untagged traffic on OpenFlow hybrid port to be treated as protected VLAN or Configured unprotected VLAN traffic, such traffic will be subjected to IP forwarding or routing.

### Configuring VLANs

Assuming the port is configured as an OpenFlow hybrid port, the following are the configuration options to consider.

**Case 1:** When a port is added as untagged in unprotected VLAN making the port configured unprotected VLAN.

Configuration will be accepted and untagged traffic on the port 2/1, for example, will be forwarded as per the OpenFlow rule if matching rule is present. If Matching OpenFlow rule is not present, untagged traffic on port 2/1 will be routed as per routing table. If route is not present, untagged traffic will be processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#vlan 300
device(config-vlan-300)#untagged ethernet 2/1
```

**Case 2:** When a port is added as untagged in a protected VLAN.

Configuration will be accepted and untagged traffic on port 2/1, for example, will be forwarded as per route table.

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#openflow protected-vlans 400
device(config-if-e10000-2/1)#vlan 400
device(config-vlan-400)#untagged ethernet 2/1
```

**Case 3:** When a port is removed from configured unprotected VLAN.

Configuration will be accepted and untagged traffic on port 2/1 will be forwarded as per OpenFlow rule, if matching rule is present. If matching OpenFlow rule is not present, untagged traffic will be processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-vlan-300)#no untagged ethernet 2/1
```

**Case 4:** When a port is removed as untagged from protected VLAN.

Configuration will be accepted and untagged traffic on port 2/1 will be dropped. Configuration will be accepted and untagged traffic on port 2/1 will be forwarded as per OpenFlow rule, if matching rule is present. If matching OpenFlow rule is not present, untagged traffic will be processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-vlan-400)#no untagged ethernet 2/1
```

**Case 5:** An untagged configured unprotected VLAN is configured as protected VLAN on a port.

Configuration will be accepted and untagged traffic on port 2/1 will be forwarded as per the route table.

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#vlan 300
device(config-vlan-300)#untagged ethernet 2/1
device(config-if-e10000-2/1)#openflow protected-vlans 300
```

**Case 6:** An untagged protected VLAN is removed from the port making it untagged configured unprotected VLAN.

It will behave as in case 1.

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#openflow protected-vlans 400
device(config-if-e10000-2/1)#vlan 400
device(config-vlan-400)#untagged ethernet 2/1
device(config-if-e10000-2/1)#no openflow protected-vlans 400
```

**Case 7:** Configured untagged unprotected VLAN is deleted.

Configuration will be accepted and untagged traffic on port 2/1 will be forwarded as per OpenFlow rule if matching rule is present. If matching OpenFlow rule is not present, untagged traffic will be processed according to the default OpenFlow rule (drop or send to controller).

```
device(config)#no vlan 300
```

**Case 8:** Untagged protected VLAN is deleted.

It will behave as in case 4.

```
device(config)#no vlan 300
```

### ***Untag match flow behavior***

If both Input port and output port belong to the same untagged VLAN, the egress packet would go out as untagged. When both input port and output port belong to different untagged VLANs, the egress packet would go as tagged.

If an OpenFlow hybrid port is already a member of untagged non-default VLAN, flow with matching VLAN as untagged non-default VLAN will be rejected. But the system will support the flow with match VLAN as 0xFFFF and a flow with Pop VLAN and same OpenFlow hybrid port in the action.

In such a scenario, if you want to send out the packet as untagged it can be pushed the VLAN matching the untagged VLAN of egress port.

You may also push some random VLAN for a flow matching VLAN 0.

***Assumptions and limitations***

1. At any time only one of the VLAN (protected VLAN or configured unprotected ) can be made as untagged member of an OpenFlow hybrid port.
2. It is supported only on OpenFlow hybrid port.
3. When protected VLAN or configured unprotected VLAN is configured as untagged VLAN, flow matching untagged VLAN cannot be accepted.
4. Normal action for untagged traffic is not supported.

## Group table

Group table introduces the ability to add support for port group abstraction for multi-pathing. This enables OpenFlow to represent a set of ports as a single entity for forwarding packets.

Group table supports the following group types.

- All - Executes all the buckets in the group; mostly used for flooding and multicasting.
- Indirect - Executes one defined bucket in the group. The action taken by this group type is sending packets to the next hop.
- Select - Executes for one bucket in the group. The action bucket is chosen by a switch-defined algorithm, such as round robin or hashing (for example, load sharing).
- Fast failover - Executes the first live bucket, such as redundancy.

A group table consists of group entries. The counters in the following table are available in a group entry.

**TABLE 27** Group entry counters

Counter	Description
Group Identifier	A 32-bit unsigned integer uniquely identifying the group
Group type	Determines group semantics
Counter	Number of packets processed by a group
Action bucket	Ordered list of action buckets, where each action bucket contains a set of actions to execute and associated parameters

## Scaling group numbers

Show OpenFlow groups gives the maximum number of actions in a bucket, the maximum number of buckets in a group and the maximum number of groups for scaling the group in OpenFlow.

```
device(config)# show openflow groups group-id
```



### ***On Brocade MLX Series and Brocade NetIron XMR devices***

- The maximum number of actions in a bucket is one.
- The maximum number of buckets in a group is 64.
- The maximum number of groups is 512.

### ***On Brocade NetIron CER and Brocade NetIron CES devices***

- The maximum number of actions in a bucket is one.
- The maximum number of buckets in a group is 64, for group SELECT, it is 12.
- The maximum number of groups is 512, for group SELECT, it is 64.

## **Considerations and limitations for group tables**

You must take into account the following when you configure group tables for OpenFlow flows.

### ***For configuring group table***

- Brocade devices support all group types in the OpenFlow 1.3 specification.
- The only action allowed in an action bucket is an output port, other supported actions can be included before group table.
- Each action bucket can have only one output port.
- Each OpenFlow port can be a part of any number of groups.
- A group entry can include ports from different slots and ports with different speed.
- Group tables are not impacted based on the OpenFlow type on interface (Layer 2 or Layer 3 or Layer23 and hybrid interfaces).
- In order to disabling OpenFlow on interfaces, the interface must be removed from any group entry first.

### ***Limitations***

For configuring OpenFlow, consider the following limitations.

- Watch\_group is not supported in Fast failover group type.
- PBR or Transparent VLAN flooding cannot be configured along with the group table, when OpenFlow 1.3 is enabled and vice versa.

Following are the additional limitations for a specific group type.

### ***For group All***

To multicast flow matching traffic to all action buckets, all action buckets will be executed every time for the group All.

- Packet is replicated for the output port in each bucket. Only one packet is processed for each bucket of the group.

### ***For group Indirect***

It executes one defined action bucket in a group. Only one action bucket can exist and it will be executed every time.

- Group Indirect supports one and only one bucket in each group entry.

### ***For group Select***

To load balance flow-matching traffic to all action buckets, one of the action buckets is chosen each time for the group Select.

- The only action allowed is output port.
- Weighted load balancing for group Select is not supported.
- Group chaining is not supported.
- Individual bucket statistics is not supported.

### ***On Brocade NetIron CER and Brocade NetIron CES devices***

These are the additional limitations for group Select on these Brocade devices.

- A port can be part of only one group.
- All the ports in a group should be of same speed.
- Once a port has become the part of the group, then this physical port cannot be used in any flow as input port (if the port type is Layer 2 ).
- Once a port has become the part of the group, then this physical port cannot be used in any flow as output port.
- If a port is already part of some flow, then group addition and modification containing that port in a bucket will be rejected .
- If an Layer 2 port is part of group Select, then generic flows will not be accepted.

### ***For group Fast failover***

It executes the first live bucket. Each action bucket associated with a specific port or group determines the liveness of the bucket.

- The buckets are selected in the defined sequence.
- On a stack unit Failover, traffic convergence will take up to 2.5 to 3 seconds.
- If no buckets are live, packets are dropped.

## **Supporting flow formats for group**

These are the supported flow formats on Brocade devices.

**On Brocade MLX Series and Brocade NetIron XMR devices:**

These devices only support one group in a flow. Flow can have multiple output ports. These are the formats supported for these devices.

o/p port: Output port

- [match ] [modify actions] [ group\_id ]
- [match ] [modify actions] [o/p port\_1] [ group\_id ]
- [match ] [modify actions] [o/p port\_1] [o/p port\_2].....[o/p port\_n] [ group\_id ]

Order of group\_id and output ports does not make a difference. These are the formats not supported in this release.

- [match ] [modify actions] [ group\_id\_1 ] [ group\_id\_2 ]
- [match ] [modify actions] [ group\_id\_1 ] [modify actions] [o/p port\_1]

Multiple groups are not supported in one flow. Separate actions is not supported for each output port and group, if group exist in a flow.

---

#### **NOTE**

If group does not exist in a flow, modify actions and output ports combination will be same as previous release.

---

#### **On Brocade NetIron CER and Brocade NetIron CES devices:**

These devices support multiple groups in one flow. This also supports separate actions for each output port and group in a flow.

- [match ] [modify actions] [ group\_id ]
- [match ] [modify actions] [o/p port\_1] [ group\_id ]
- [match ] [modify actions] [o/p port\_1] [o/p port\_2].....[o/p port\_n] [ group\_id ]
- [match ] [modify actions] [ group\_id\_1 ] [ group\_id\_2 ]
- [match ] [modify actions] [ group\_id\_1 ] [modify actions] [o/p port\_1]

## **Group events**

These are the group events supported by OpenFlow.

The following group events are handled by OpenFlow control planes resided in Management Module to update group information to program hardware for forwarding information.

- Add group
- Delete group
- Add port to the group
- Delete port from the group
- Group type modification
- Group output port is up
- Group output port is down

### ***Statistics***

Group statistics are cumulative flow statistics that use the group ID in the action list. The following statistics are supported per group.

- Reference count (flow entries)
- Packet count (limited support on different devices)
- Byte count

- Duration (second)
- Duration (nanosecond)

**For OpenFlow hybrid ports**

- Group table does not affect hybrid functionality.
- Flows within a group on the hybrid port are treated the same as other flows.
- A group can support Normal and hybrid OpenFlow port together.

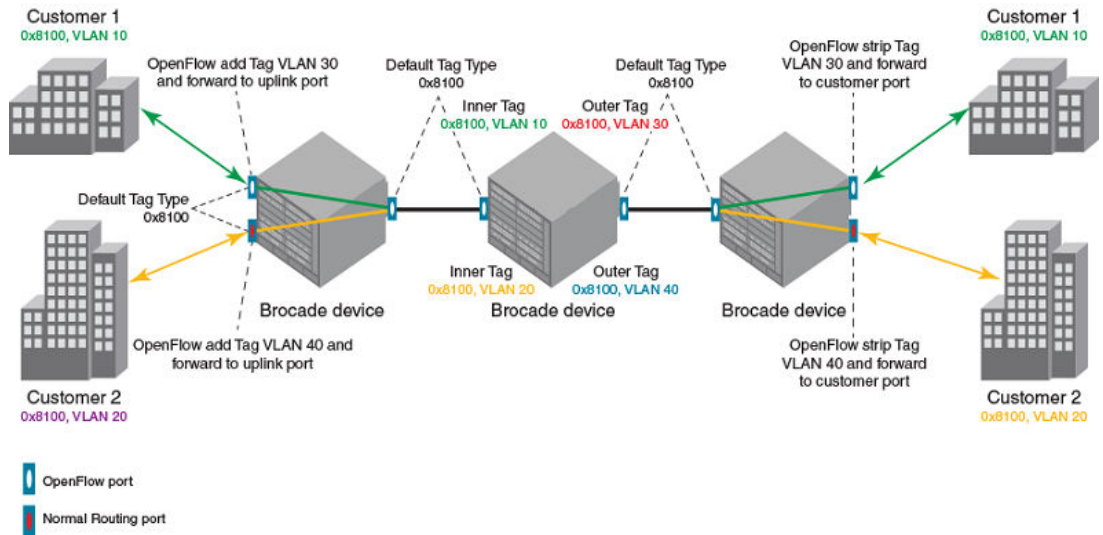
# QinQ

You can push or pop or set VLAN tags in the outgoing packets of an OpenFlow flow with QinQ support. The ingress packet can be untagged, single tagged or double tagged frames. You can use QinQ to transport multiple customer segments or VLANs across Layer 2 infrastructures.

A OpenFlow flow matches on wild card VLAN or single VLAN and it does one of the following.

- No action
- Push outer VLAN
- Pop outer VLAN
- Modify or set outer VLAN

**FIGURE 8** Transportation of VLAN or segment using QinQ



When a frame with QinQ tags arrives at the switch, the switching decision is made to match on the outer tag till the frame reaches a node configured to remove the outer tag.

On the customer ports, matching customer VLAN tag is supported. OpenFlow metering is also supported on flows with matching customer VLAN tags. Traditional features are supported for packets tagged with configured tag-type on the hybrid port.

- The customer VLANs are supported with hybrid functions on the hybrid ports with matching tag-type configuration.
- The customer packets can be forwarded by OpenFlow and traditional features.

**NOTE**

Matching fields in flow (VLAN + Layer 3 fields) will be supported on Layer 3 and Layer23 OpenFlow Port.

**NOTE**

Matching fields in flow (VLAN + Layer 2 fields) will be supported on Layer 2 and Layer23 OpenFlow Port.

On the uplink and transit port, OpenFlow can match outer VLAN tag. It supports addition, stripping or matching outer VLAN tag in Layer 2, Layer 3 and Layer23 mode. Hybrid port mode is supported for single tagged packets, but not for dual tagged packets. OpenFlow metering is also supported on flows with matching Outer VLAN tags. Traditional features are supported the packets tagged with configured tag-type on the hybrid port as well on uplink and transit ports.

## Flow validation checks

Consider the following points when you configure OpenFlow 1.3 on devices for QinQ support.

- If the flow has more than one Push, Pop or Set VLAN field, the flow is not supported.
- If the flow includes two and above Push, Pop, or Set VLAN field, it is not supported.
- Layer 3 action fields cannot be supported in conjunction with Openflow actions PUSH and POP VLAN, such a flow will be rejected. Similarly source and destination MAC modification action is not supported along with PUSH VLAN action.
- For every push VLAN action there has to be a corresponding set or modify VLAN action field.

**NOTE**

Tag Type configuration is a port level setting, so all flows on a port will recognize a packet as tagged only, if it is coming with the same Ether type as configured. Packets with other tag types will be treated as untagged.

**NOTE**

Packets with more than two tags will be dropped on a Layer 3 OpenFlow port.

**NOTE**

OpenFlow v1.3 QinQ flows with Push VLAN action is not supported on Brocade NetIron CER and Brocade NetIron CES series on Layer 2 OpenFlow port.

## QinQ action

These are the enhancements to the existing QinQ functionality.

- Push, Pop or Modify up to 2 VLAN tags per flow
- Inner VLAN tag ether type can be modified

With OpenFlow 1.3 QinQ support, you will be able to Push, Pop or Set up to 2 VLAN tags in the outgoing packets of an OpenFlow flow. The ingress packet can be untagged, single tagged or double tagged frames. You can use QinQ to transport multiple customer segments or VLANs across Layer 2 infrastructures.

**TABLE 28** Open flow actions

OpenFlow action	Input traffic			
	Untagged	Single tagged	Double tagged	More than 2 tags
No Action	Egress traffic goes as untagged. (NO OP)	Single VLAN in ingress packet should be preserved when sent out in egress direction.  (NO OP)	Two VLANs in ingress packet should be preserved when sent out in egress direction.  (NO OP)	More than two VLAN tags in ingress packet should be preserved when sent out in egress direction.  (NO OP)
Push 2 VLANs	It goes out as dual tagged packet at egress.	It goes out with 3 tags at egress.	It goes out with 4 tags at egress.	It goes out with 5 tags at egress.
Pop 2 VLANs	Should go as untagged in egress direction. (NO OP)	Should go as untagged in egress direction as ingress VLAN tag is stripped.	Should go as untagged in egress direction.	It should go out as single tagged(with outer and inner VLAN removed) packet at egress.
Modify/Set 2 VLANs	It should go out as dual tagged packet at egress	It should go out as dual tagged packet at egress.	It should go out as dual tagged packet at egress.	It should go out as triple tagged packet at egress.

Inner VLAN ethertype can be modified to any user defined values.

#### Setting tag-type for Inner tag

The following tag-type command is modified to include an option to configure the tag-type for inner tags.

```

device(config)#tag-type ?
  HEX      etype in hex (Default: 0x8100)
  isid     Specify Etype for I-Tagged packets
  inner    Specify Etype for Inner Tag

device(config)#tag-type inner ?
  HEX      etype in hex (Default: 0x8100)

device(config)#tag-type inner <inner-tag-type> ?
  ethernet Ethernet port
  pos      POS port

device(config)#tag-type inner <inner-tag-type> e
  ethernet Ethernet port

device(config)#tag-type inner <inner-tag-type> ethernet ?
  SLOT/PORT Interface number

device(config)#tag-type inner <inner-tag-type> ethernet <slot/port> ?
  ethernet Ethernet port
  to      To an end range
  <cr>

device(config)#tag-type inner <inner-tag-type> ethernet <slot/port>

```

**Show tag type**

```
device(config)#show tag-type
```

```
VLAN TAG ETYPE
```

```
-----
```

```
Global Tag-type 0x8100
```

Port	Tag-Type	Inner-Tag-Type
2/3	0x88a8	0x8100

```
I-TAG ETYPE
```

```
-----
```

```
Global I-TAG Tag-type 0x88e7
```

These are the examples for LP.

**Show Openflow flows (2 Push Action)**

New fields have been added for Outer and Inner VLAN id.

```
device(config)#show openflow flows
```

```
.....
```

```
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
```

```
Rule:
```

```
  In Port:      e2/1
  In Vlan:      Tagged[1210]
```

```
Action: FORWARD
```

```
  Out Port:    e2/1, Push-vlan-tag 1: 0x8100, Vlan id: 3
  Push-vlan-tag 2: 0x8100, Vlan id: 4
```

```
  FID: -N/A-, MVID: -N/A-
```

```
Hardware Information:
```

```
Port: 2/1  PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63
```

```
Packets: 0
```

```
Statistics:
```

```
  Total Pkts: 0
  Total Bytes: 0
```

**Show Openflow flows (2 Pop Action)**

```
device(config)#show openflow flows
```

```
.....
```

```
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
```

```
Rule:
```

```
  In Port:      e2/1
  In Vlan:      Tagged[1210]
```

```
Action: FORWARD
```

```
  Out Port:    e2/1, Pop-vlan-tag 1
  Pop-vlan-tag 2
```

```
  FID: -N/A-, MVID: -N/A-
```

```
Hardware Information:
```

```
Port: 2/1  PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63
```

```
Packets: 0
```

```
Statistics:
```

```
  Total Pkts: 0
  Total Bytes: 0
```

**Show Openflow flows (2 Set Action)**

```
device(config)#show openflow flows
```

```
.....
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
Rule:
  In Port:      e2/1
  In Vlan:      Tagged[1210]
Action: FORWARD
  Out Port:     e2/1, Vlan id 1: 3
                Vlan id 2: 3

  FID: -N/A-, MVID: -N/A-
Hardware Information:
Port: 2/1  PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63
Packets: 0
Statistics:
  Total Pkts: 0
  Total Bytes: 0

Vlan id: 3
```

**Show Openflow flows (Push-vlan-tag and push-in-vlan-tag)**

```
device(config)#show openflow flows
```

```
.....
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
Rule:
In Port: e2/1
In Vlan: Tagged[1210]
Action: FORWARD
Out Port: e2/1, Push-vlan-tag 1: 0x8100, Vlan id: 3
Push-vlan-tag 2: 0x8100, Vlan id: 4
FID: -N/A-, MVID: -N/A-
Hardware Information:
Port: 2/1 PPCR Id : 3, CAM Index: 0x000576aa (L4) PRAM Index: 0x0003ff63
Packets: 0
Statistics:
Total Pkts: 0
Total Bytes: 0
```



These are the outputs for MP.

### Show Openflow flows (2 Push tag)

```
device(config)#show openflow flows
.....
Total Number of data packets sent to controller:          0
Total Number of data bytes sent to controller :          0

Total Number of Flows: 3
  Total Number of Port based Flows: 3
  Total Number of L2 Generic Flows: 0
  Total Number of L3 Generic Flows: 0
  Total Number of L2+L3 Generic Flows: 0
  Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 3
  Total Number of Hardware entries for Port flow: 3
  Total Number of Hardware entries for Generic flow: 0

Total Number of Openflow interfaces: 4
  Total Number of L2 interfaces: 0
  Total Number of L3 interfaces: 2
  Total Number of L23 interfaces: 2

Flow ID: 5 Priority: 32768 Status: Active
Rule:
  In Port:          e12/1
  Instructions: Apply-Actions
  Action: FORWARD
  Out Port:  e12/2, Push-vlan-tag:          0x00008100 id: 100,
Vlan PCP: 1
  Push-in-vlan-tag:          0x00008100 id: 200,
In Vlan PCP: 2
  Statistics:
    Total Pkts: 0
    Total Bytes: 0
```

### Show Openflow flows (2 Pop tag)

```
device(config)#show openflow flows
.....
Flow ID: 6 Priority: 32768 Status: Active
Rule:
  In Port:          e12/2
  Instructions: Apply-Actions
  Action: FORWARD
  Out Port:  e12/1, Pop-vlan-tag
  Pop-in-vlan-tag

Statistics:
  Total Pkts: 0
  Total Bytes: 0
```

### Show Openflow flows (2 Set Action)

```
device(config)#show openflow flows
.....
Flow ID: 7 Priority: 32768 Status: Active
Rule:
  In Port:          e12/1
  In Vlan:          Tagged[100]
  Instructions: Apply-Actions
  Action: FORWARD
  Out Port:  e12/2, Vlan: id: 500, Vlan PCP: 5
  In Vlan: id: 600, In Vlan PCP: 6

Statistics:
  Total Pkts: 0
  Total Bytes: 0
```

**Limitations**

1. Push, Pop or Modify action cannot be combined in one single command.
2. For Multi-output port case, all output ports should have the same VLAN action.
3. Layer 3 field, source MAC and Destination MAC modification cannot be supported along with Push VLAN action.
4. Push VLAN action is not supported on Layer 2 OpenFlow port on Gen1 Card.
5. Inner VLAN PCP modification is not supported when 2 VLANs are pushed or modified in the OpenFlow action.
6. VLAN preservation is not supported on Gen1cards, thus modify or pop action is applicable only on outermost VLAN.

# Enqueue

The controller is able to set up and configure queues and then map flows to a specific queue. The queue configuration sets the queue ID for a packet and determines the queue to be used for scheduling and forwarding the packet.

Queue configuration takes place outside the OpenFlow protocol based on weights for a particular queue using Weighted Round Robin (WRR) scheduling.

```
device(config-if-e10000-2/5)# show qos scheduler ethernet 2/5
Port | Scheme  Type  Pri7  Pri6  Pri5  Pri4  Pri3  Pri2  Pri1  Pri0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
  2/5 | weighted Weight 10   10   20   10   10   20   10   10
```

There are two distinct parts that form the enqueue mechanism:

- Configuration
- Flow-queue mapping or forwarding

Assuming that a queue is already configured, you can associate a flow with an OFPAT\_ENQUEUE action which forwards the packet through the specific queue on a port. Note that an enqueue action will override any TOS or VLAN\_PCP-related behavior that is potentially defined in the flow, but the packet will not be changed or modified due to an enqueue. Brocade devices support a total of 8 queues per port.

In case of stacking, queue 7 is reserved for stacking messages. Any queue set to 7 will be reclassified to queue 6. When there is no stacking, the standalone queue set to 7 will remain as 7.

## Use case - OpenFlow meter and enqueue

QoS is usually implemented to provide appropriate levels of service to support Service Level Agreements(SLAs). Here is an example, in which you have the ability to meter and determine customer traffic according to the bandwidth guaranteed provided to the customer via a combination of OpenFlow 1.0 or 1.3 actions. The policing must be fine grained and flexible enough as supported by OpenFlow match semantics. For instance, the match criteria for rate limiting one application may be based on VLAN tag and, for other application, it may be based on the layer 4 UDP or TCP port. The confirm action will set appropriate queue-id for the packets, while exceed action may cause the traffic to be dropped in case of congestion or remarked to lower priority and with a different queue-id. When the packet is forwarded to a port using the output action, the queue- id will determine which queue attached to this port is used for scheduling and forwarding the packet.

## Configuring OpenFlow Enqueue

Queue configuration takes place outside the OpenFlow protocol, either through a command line tool or through an external dedicated configuration protocol.

The minimum guaranteed bandwidth is configured through assignment of weights for a particular Queue (with WRR scheduling).

Brocade NetIron CER Series and Brocade NetIron CES Series requires extended QoS mode configuration to support the OpenFlow enqueue feature on a Layer 2 OpenFlow port.

Do the following to configure OpenFlow enqueue.

1. Enable queue statistics at global level.
 

```
device (config) # statistics
device (config-statistics # tm-voq-collection
```
2. Configure WRR Scheduling and weights for the queues at the egress.
 

```
device(config-if-e10000-2/5)# qos scheduler weighted 10 10 20 10 10 20 10 10
```
3. Configure Shaper configuration for the queues at the egress port (configuring maximum rate).
 

```
device(config-if-e10000-2/5)#qos shaper priority 3 3000
```
4. Disable encode policy map at egress port.
 

```
device(config-if-e10000-2/5)#qos pcq encode-policy off
device(config-if-e10000-2/5)#qos dscp encode-policy off
```
5. Configure priority queues from 8 to 4 or vice versa.
 

```
device(config)#system-max-tm-queues 4
```

The queues are now configured for forwarding actions. After the queues have been configured, flows can be mapped to queues and packets will be forwarded through them.

## Limitations

These are the limitations for the enqueue.

- A flow can have a maximum of one queue id which is applicable for all output port in the action list.
- OpenFlow flows with action as Set IP TOS or Set VLAN PCP cannot be supported simultaneously with enqueue configuration. Such configuration will be rejected.
- QoS functionality of hybrid traffic flowing through these ports will be affected.
- OpenFlow queue statistics is actually retrieved from existing TM statistics. The limitation of TM statistics applies for OpenFlow queue statistics as well.
- OpenFlow queue statistics is not supported for multicast flows.

## Queue statistics

The following are the statistics supported per Queue.

**TABLE 29** Statistics supported on Brocade devices

Queue statistics	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Number of transmitted packets	Yes	No
Number of transmitted bytes	Yes	No
Transmit overrun error, number of packets dropped due to overrun	No	No

**TABLE 29** Statistics supported on Brocade devices (Continued)

Queue statistics	Brocade NetIron MLX Series	Brocade NetIron CER Series
	Brocade NetIron XMR Series	Brocade NetIron CES Series
Duration (seconds), time queue has been alive	No	No
Duration (nanoseconds), time queue has been alive	No	No

**NOTE**

A flow can have a maximum of one queue-id and applicable for all output ports in the action list.

To display queue level statistics, use this command as following.

```
device(config)# show np qos statistics ethernet 2/1
Port 2/1
  Ingress counters:
    COS 0: packets 1918620           bytes 168838560
    COS 1: packets 0                 bytes 0
    COS 2: packets 0                 bytes 0
    COS 3: packets 0                 bytes 0
    COS 4: packets 0                 bytes 0
    COS 5: packets 0                 bytes 0
    COS 6: packets 0                 bytes 0
    COS 7: packets 0                 bytes 0
```

## Displaying OpenFlow queues

The **show openflow queue** command displays the queues attached to egress ports.

Ensure that OpenFlow queuing is configured on the device. The user can associate a flow with an **OFFPAT\_ENQUEUE** action which forwards the packet through the specific queue on a port.

1. Return to global configuration mode.
2. Enter the **show openflow queues** command and specify the queue ids.

```
device(config)# show openflow queues 2/5

Openflow Port    2/5
  Queue 0
    Min Rate: 107374200 bps           Max Rate: 858993600 bps
    Tx Packets: 0
    Tx Bytes: 0
Openflow Port    2/5
  Queue 1
    Min Rate: 107374200 bps           Max Rate: 858993600 bps
    Tx Packets: 0
    Tx Bytes: 0
Openflow Port    2/5
  Queue 2
    Min Rate: 214748400 bps           Max Rate: 858993600 bps
    Tx Packets: 0
    Tx Bytes: 0
Openflow Port    2/5
  Queue 3
    Min Rate: 107374200 bps           Max Rate: 858993600 bps
    Tx Packets: 0
    Tx Bytes: 0
Openflow Port    2/5
  Queue 4
    Min Rate: 107374200 bps           Max Rate: 858993600 bps
    Tx Packets: 0
    Tx Bytes: 0
Openflow Port    2/5
  Queue 5
    Min Rate: 214748400 bps           Max Rate: 858993600 bps
```

```

Tx Packets: 0
Tx Bytes: 0
Openflow Port 2/5
Queue 6
Min Rate: 107374200 bps           Max Rate: 858993600 bps
Tx Packets: 0
Tx Bytes: 0
Openflow Port 2/5
Queue 7
Min Rate: 107374200 bps           Max Rate: 858993600 bps
Tx Packets: 0
Tx Bytes: 0

```

The output displays the minimum and maximum traffic rates in each interface.

## Metering

Per-flow metering measures and controls the rate of packets for each flow entry. Per-flow meters enable OpenFlow to implement simple QoS operations, such as rate-limiting, and can be combined with per-port queues to implement complex QoS frameworks, such as DiffServ.

Meters are attached directly to flow entries. Each meter can have one or more meter bands. Each meter band specifies the rate of the band applies and the way packets are processed (DROP or DIFFSERV). OpenFlow metering operation is similar to ingress rate limiting in a QoS operation.

---

### NOTE

Metering is not supported on Brocade NetIron CER Series and Brocade NetIron CES Series.

---

A meter table consists of meter entries. The counters in the following table are available in the meter entry.

**TABLE 30** Meter entry

Counter	Description
Meter Identifier	A 32-bit unsigned integer uniquely identifying the meter
Meter band	A list of meter bands, where each meter band specifies the rate of the band and the way to process the packet. Rate and burst size are based on the line rate of the data traffic in contrast to the information rate.
Counter	Number of packets processed by a meter

Packets are processed by a single meter band based on the current measured meter rate. The meter applies the meter band with the highest configured rate that is lower than the current measured rate. If the current rate is lower than any specified meter band rate, no meter band is applied.

**TABLE 31** Meter bands supported on Brocade devices

Meter bands	Supported
DROP	Yes
DSCP_REMARK	Yes
EXPERIMENTER	No

Each band type contains the following meter configuration parameters from the controller:

- Rate value in kbps
- Rate value in packets per second
- Burst size
- Statistics collection

**TABLE 32** Meter configuration parameters

Configuration flags	Supported
OFPMF_KBPS	Yes
OFPMF_PKTPTS	No
OFPMF_BURST	Yes
OFPMF_STATS	Yes

The metering system supports the features in the following table.

**TABLE 33** Metering capabilities supported for metering features

Feature	Brocade NetIron MLX Series
	Brocade NetIron XMR Series
Maximum meter available in the system	3959
Band types (bitmap)	DROP, DSCP_REMARK
Capabilities (bitmap)	KBPS, BURST, STATS
Maximum number of band per meter	1
Maximum color value	3 (RED, YELLOW, GREEN)

## Meter statistics

The following statistics are supported per meter:

- Flow count (number of flows associated with the meter)
- Input byte count (cumulative byte count on all associated flows)
- Duration (second)
- Duration (nanosecond)

The flow and the byte count calculate all packets processed by the meter. The duration fields indicate the elapsed time for which the meter has been installed on the device.

The byte band count counter is supported for the meter band type.

The byte band count presents the total numbers for all bytes processed by the band.

**TABLE 34** Meter band statistics

Band Type	Supported
DROP	Yes
DSCP_REMARK	No

## Limitations

The following limitations apply to the Brocade devices for metering:

### ***On Brocade MLX Series and Brocade NetIron XMR devices***

- A Meter cannot be applied with DSCP remark meter band alone.
- A maximum of 3959 non-priority based meters can be created in this device provided other application do not use it, as it is a shared resource and a maximum of 989 priority-based meters can be supported. These are shared resources with other application.
- When flows associated to a meter are all from different device in a device, applying exact meter rate is not possible.
- Packet count is not supported for individual meter and meter bands.
- Meters are not supported on flows with action as DROP or CONTROLLER.

### ***On Brocade NetIron CER and Brocade NetIron CES devices***

- If a flow is associated with a meter, flow statistics is not supported. The meter statistics is not substituted for flow statistics as the meter can be associated with more than one flow.
- Meter and counter share same hardware resources. The maximum resources available are 2K. These can be used either as meters or as counters (flow statistics).
- These 2K resources are shared with all other applications.
- If you program flows without a meter, it will, by default, consume a counter or meter resource. Brocade recommends you configure all meters before configuring any flows to the system.

### ***Meter band***

The following limitations apply to the meter bands:

- The minimum burst size for DSCP or DROP band is 10 Kilobits.
- The maximum rate for DROP or DSCP is 100,000,000 kbps; the minimum is 0 kbps.
- The DSCP band rate cannot be greater than DROP band rate.
- The precedence level for DSCP band type should be from 0 through 63.

## Displaying OpenFlow meters

A meter measures the rate of packets assigned to it and enables controlling the rate of those packets.

Return to global configuration mode.

The hardware resources are shared between OpenFlow and other features, so these resources are allocated on a first-come-first-serve basis.

Enter the **show openflow meters** command to showing all the meters in a flow for MP.

```
device(config)# show openflow meters 2
Meter id: 2

Transaction id:      1438
Meter Flags:         KBPS BURST STATS
Flow Count:          0
Number of bands:     2
In packet count:     -NA-
In byte count:       0

Band Type:          DSCP-REMARK

Rate:                750000
Burst size:          1500          kb
```

```

Prec level:          1
In packet band count: -NA-
In byte band count:  0

Band Type:    DROP

Rate:          1000000
Burst size:    2000      kb
In packet band count: -NA-
In byte band count:  0

----
Total no. of entries printed: 1

```

The following example output shows a specific meter for LP.

```

device(config)# show openflow meters 1
Meter id: 1023

Meter Flags:          KBPS BURST
Number of bands:     2
RL Class Index:      33      33
In packet count:     -NA-
In byte count:       0

Band Type:    DROP

Rate:          3000      Adjusted rate:2996
Burst size:    1250      kb
In packet band count: -NA-
In byte band count:  0

Band Type:    DSCP-REMARK

Rate:          1700      Adjusted rate:1693
Burst size:    1250      kb
Prec level:    27
In packet band count: -NA-
In byte band count:  0

```

Meter implementation does not address any vendor specific proprietary messages.

## TTL support

TTL-based actions are supported in the OpenFlow flows.

The system supports decrement TTL-based actions in OpenFlow flow for Brocade MLX Series and Brocade NetIron XMR devices and Brocade NetIron CER and Brocade NetIron CES devices. Decrement TTL action is supported if flow is having match with ethertype 0x800. It only supports Set TTL based action for Brocade NetIron CER and Brocade NetIron CES devices.

**show openflow** flow will have a field for SET TTL or DEC TTL in the flow action display.



## Forwarding port and controller for matching ARP

If the flow has ARP as ETH\_TYPE match, action can have output port or group along with send to controller option.

The following example shows openFlow flows output on MP.

```
ovs-ofctl add-flow tcp:10.25.123.245 --protocols=OpenFlow13 "in_port=145
dl_type=0x0806 actions=output:152,controller"
```

```
Flow ID: 6 Priority: 32768 Status: Active
Rule:
  In Port:      e4/1
  Ether type:   0x00000806
  Instructions: Apply-Actions
    Action: FORWARD
      Out Port:  e4/8

      Out Port: send to controller
```

It will be supported on all Layer 2, Layer 3 and Layer23 OpenFlow ports. Packets sent to the controller (CPU) will be ratelimited to maximum 500 packets/sec. This number will be shared between different flows having action as 'send to controller'. Along with ARP, other Layer 2 fields can be supported in the match criteria.

---

### NOTE

Only Layer 2 and Layer23 port will be supported on Brocade NetIron CER Series and Brocade NetIron CES Series.

---

### Limitations

- Only ETH\_TYPE as 0x0806 will be supported for such type of flows.
- Flow modify or set action for this flow will be rejected. Action in such flows can have only output\_port, out\_group or controller. It cannot change the incoming traffic for such flows.

## sFlow support on OpenFlow ports

To support sFlow sampling on OpenFlow ports, the existing sFlow commands can be used on the OpenFlow port.

All traffic on a protected VLAN, unprotected VLAN, and OpenFlow received on an OpenFlow port can be monitored together.

To enable sFlow forwarding on OpenFlow interfaces, you must do the following:

- Globally enable sFlow
- Enable sFlow forwarding on individual interfaces

```
device(config)# sflow enable
device(config)# interface ethernet 1/1 to 1/8
device(config-mif-1/1-1/8)# sflow forwarding
```

To change the sampling rate on an individual port, enter a command such as the following at the configuration level for the port.

```
device(config-if-e10000-1/1)# sflow sample 8192
```

To specify sFlow collectors, enter a command such as the following.

```
device(config)# sflow destination 10.10.10.1  
device(config)# sflow destination ipv6 10:10::10:10
```

## TLS1.2 support for NetIron devices

SSL or TLS is the Transport Layer Security protocol that provides a secure communications data channel for applications in NetIron devices such as HTTPS web server, syslog, TACACS+, and OpenFlow communications.

NetIron devices support the SSL V3.0, TLS1.0, and TLS1.1 protocols. To support the TLS1.2 protocol, the cryptographic engine has been updated.

The TLS1.2 protocol is functional in both server mode and client mode according to RFC 5246.

### TLS1.2 in server mode

The HTTPS server communicates with the HTTPS clients or web browsers using the TLS1.2 protocol supporting all server functionality, such as user authentication.

The following popular web browsers are supported:

- Microsoft Internet Explorer
- Google Chrome
- Firefox

### TLS1.2 in client mode

The NetIron device is a client and communicates with the remote server using the TLS1.2 protocol.

TLS1.2 in client mode supports the client certificate because the peer TLS1.2 server may request a certificate from the client.

The syslog and TACACS+ protection is enabled in the Common Criteria mode of operation.

# SDN Commands

---

• show openflow.....	92
• show openflow controller.....	94
• openflow controller source-interface.....	95
• show openflow interfaces.....	96
• show openflow flows.....	98
• openflow enable .....	100
• clear openflow .....	101
• show openflow groups.....	102
• show openflow queues.....	103
• show openflow meters.....	105
• clear statistics openflow .....	107
• openflow hello-reply disable.....	108

## show openflow

Displays the configured OpenFlow parameters.

**Syntax** `show openflow`

**Modes** EXEC and Privileged EXEC mode  
Global configuration mode

**Command Output** The `show openflow` command displays the following information:

Output field	Description
Administrative Status	Enable or disable status
Controller Type	OpenFlow 1.0 or OpenFlow 1.3 controller
Controller	Number of controllers

### Examples

```
device#show openflow

Administrative Status:      Enabled
Controller Type:           OFV 130
Number of Controllers: 4

Controller 1:
Connection Mode:           passive, TCP
Listening Address:         0.0.0.0
Connection Port:           6633
Connection Status:         TCP_LISTENING
Role:                       Equal
Asynchronous Configuration: Packet-in (no-match|action|invalid-ttl)
                             Port-status (add|delete|modify)
                             Flow-removed (idle-timeout|hard-timeout|delete|grp-
delete)

Controller 2:
Connection Mode:           active, TCP
Controller Address:         10.25.128.243
Connection Port:           2001
Connection Status:         OPENFLOW_ESABLISHED
Role:                       Master
Asynchronous Configuration: Packet-in (no-match|action|invalid-ttl)
                             Port-status (add|delete|modify)
                             Flow-removed (idle-timeout|hard-timeout|delete|grp-
delete)

Controller 3:
Connection Mode:           active, TCP
Controller Address:         10.25.128.242
Connection Port:           6633
Connection Status:         OPENFLOW_ESABLISHED
Role:                       Slave
Asynchronous Configuration: Port-status (add|delete|modify)

Controller 4:
Connection Mode:           active, TCP
Controller Address:         10.25.128.250
Connection Port:           2002
Connection Status:         OPENFLOW_ESABLISHED
Role:                       Slave
Asynchronous Configuration: Port-status (add|delete|modify)

Match Capability:
Port, Destination MAC, Vlan, Vlan PCP
Openflow Enabled Ports:    e1/1 e1/2
```

History	Release version	Command history
	NI05.5.00	This command was introduced.
	NI05.7.00	This command was modified for OpenFlow 1.3

# show openflow controller

Displays the controller information in a flow.

**Syntax** `show openflow controller`

**Modes** EXEC and Privileged EXEC mode  
Global configuration mode

**Command Output** The `show openflow controller` command displays the following information:

Output field	Description
Mode	Gives the active and passive connection of the controller.
IP address	IP address of the port
Port	Port number
Status	After the connection and OpenFlow handshake, the controller gives the role of OpenFlow channel.
Role	Equal, Master and Slave role for the controller.

**Examples**

```
device# show openflow controller
-----
Contlr Mode  TCP/SSL IP-address  Port  Status  Role
-----
1  (Equal)  passive  TCP  0.0.0.0  6633  TCP_LISTENING
2  (Master)  active  TCP  10.25.128.179  6633  OPENFLOW_ESABLISHED
3  (Slave)  active  TCP  10.25.128.177  6633  OPENFLOW_ESABLISHED
3  (Equal)  active  TCP  10.25.128.165  6633  OPENFLOW_ESABLISHED
```

**History**

Release version	Command history
NI05.5.00	This command was introduced.
NI05.7.00	This command was modified to give information about the role of the controller.

# openflow controller source-interface

Configures a source-interface for the connection from the device to the controller.

**Syntax** **openflow controller source-interface** { **ethernet** *slot/port* | **loopback** *number* | **ve** *number* } **force-reconnect**

**no openflow controller source-interface** { **ethernet** *slot/port* | **loopback** *number* | **ve** *number* } **force-reconnect**

**Command Default** The CLI command is applicable only when the device is in active mode. The device initiates connection to the remote OpenFlow controller.

**Parameters** **ethernet** *slot port*

Gives information about a particular slot and port in an internet

**loopback** *number*

Specifies a loopback interface.

**ve** *number*

Specifies a virtual interface.

**force-reconnect**

Forces the existing connections to use the newly configured source-interface.

**Modes** Privileged EXEC mode

**Usage Guidelines** When adding a new controller to the device, a connection will be attempted to the controller IP address using the configured source-interface. If the source-interface has no IP address configured or the interface is down, the syslog messages will be generated and a connection attempt will be made again in 15 seconds.

**Examples** To see the source-interface, use this command.

```
device(config)#openflow controller ?
ip-address      Set the Controller IPv4 address
passive        Configure passive connection mode
source-interface Set the Source Interface to be used for controller
connections
```

If a new controller is added after this, routing table will be used to connect to the controller.

```
Device(config)#openflow controller source-interface ?
ethernet      Ethernet interface
loopback      Loopback interface
ve            Virtual Ethernet interface
```

For a specified ethernet interface, use this command.

```
device(config)#openflow controller source-interface ethernet 2/2?
force-reconnect Force the existing connections to use the newly configured
source-interface
```

## History

Release version	Command history
5.8.00	This command was introduced.

# show openflow interfaces

Displays the information about the interfaces in a OpenFlow flow.

**Syntax** `show openflow interfaces`

**Modes** User EXEC mode  
 Privileged EXEC mode  
 Global configuration mode

**Usage Guidelines**

**Command Output** The `show openflow interfaces` command displays the following information:

Output field	Description
Port	Port Number
Link	Link status
Speed	Configured speed
Tag	Tag status
Mac Address	MAC address of the port
Mode	Gives the information about the layers

**Examples**

```
device# openflow enable layer3 hybrid
device# show openflow interfaces
```

Total number of Openflow interfaces: 5

Port	Link	Speed	Tag	MAC	OF-portid	Name	Mode
1/1	Up	1G	Yes	000c.dbf5.bd00	1		Layer2
1/2	Up	1G	Yes	000c.dbf5.bd01	2		Layer2
1/3	Up	1G	Yes	000c.dbf5.bd01	3		Hybrid-Layer3
1/4	Up	1G	Yes	000c.dbf5.bd01	4		Hybrid-Layer3
1/5	Up	1G	Yes	000c.dbf5.bd01	5		Hybrid-Layer3



This command displays information for a particular interface on a specific slot and port..

```
device# show interface ethernet 1/1/6

GigabitEthernet1/1/6 is up, line protocol is up
  Port up for 51 minutes 53 seconds
  Hardware is GigabitEthernet, address is 748e.f8e7.d901 (bia 748e.f8e7.d901)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDI
  Member of L2 VLAN ID 100, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled, Designated protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0, mac-learning is enabled
  OpenFlow enabled, Openflow Index 1, Flow Type Layer2
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  MTU 1500 bytes, encapsulation ethernet
  300 second input rate: 3904 bits/sec, 7 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  23153 packets input, 1530094 bytes, 0 no buffer
  Received 1721 broadcasts, 21432 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
  Relay Agent Information option: Disabled

Egress queues:
Queue counters      Queued packets      Dropped Packets
  0                   0                    0
  1                   0                    0
  2                   0                    0
  3                   0                    0
  4                   0                    0
  5                   0                    0
  6                   0                    0
  7                   0                    0
```

#### History

Release version	Command history
NI05.4.00	This command was introduced.

## show openflow flows

Displays the flows information on the OpenFlow ports.

**Syntax** `show openflow flows`

**Modes** User EXEC mode  
Privileged EXEC mode  
Global configuration mode

**Command Output** The `show openflow flows` command displays the following information:

Output field	Description
Flow	Number of flows
Packet	Total Number of data packets trapped to be sent to controller
Byte	Total Number of data bytes trapped to be sent to controller

**Examples** This command displays the output for MP.

```
device# show openflow flows

Total Number of data packets sent to controller:          0
Total Number of data bytes sent to controller :          0

Total Number of Flows: 1
  Total Number of Port based Flows: 1
  Total Number of L2 Generic Flows: 0
  Total Number of L3 Generic Flows: 0
.....
.....

Flow ID: 1 Priority: 32768 Status: Active
  Rule:
    In Port:      e2/5
  Instructions: Apply-Actions
    Action: FORWARD
    Out Port:    e2/1
    Meter id: 1023
  Statistics:
    Total Pkts: 0
    Total Bytes: 0
```

This command displays the output for LP.

```
device# show openflow flows

Total Number of data packets trapped to be sent to controller:      0
Total Number of data bytes trapped to be sent to controller :      0

Total Number of Flows: 1

Flow Id: 1, Priority: 32768, FD Id: 0, PW Id: 1
  Rule:
    In Port:      e2/1
    Action: FORWARD
    Out Port:     e2/1, Queue: 4

    FID: -N/A-, MVID: -N/A-
    Hardware Information:
    Port: 2/1   PPCR Id : 3, CAM Index: 0x000576ac (L4)   PRAM Index: 0x0003ff5e
Packets: 0
  Statistics:
    Total Pkts: 0
    Total Bytes: 0
```

History	Release version	Command history
	NI05.5.00	This command was introduced.
	NI05.7.00	This command was modified for OpenFlow 1.3

## openflow enable

Enables or disables the OpenFlow hybrid port-mode on the port.

**Syntax** `openflow enable [ layer2 | layer3 | layer23 [hybrid-mode ] ]`

`no openflow enable [ layer2 | layer3 | layer23 [hybrid-mode ] ]`

**Parameters** `layer2`

Enables Layer 2 matching mode for flows.

`layer3`

Enables Layer 3 matching mode for flows.

`layer23 hybrid-mode`

Enables Layer 2 and Layer 3 matching mode for flows with an option for hybrid port-mode.

**Modes** Interface configuration mode

**Usage Guidelines** In interface configuration mode, this command enables Layer 2 or Layer 3 matching mode for flows with an optional enabling of hybrid port-mode.

---

### NOTE

OpenFlow must be globally enabled before the Layer 2 or Layer 3 matching modes can be specified.

---

**Examples** After OpenFlow 1.3 is enabled, the following example configures Layer 2 and Layer 3 matching mode for flows.

```
device# configure terminal
device(config)# openflow enable ofv130
device (config)# interface ethernet 1/1/1
device (config-if-1/1/1)# openflow enable layer 23
```

**History**

Release	Command History
NI5.6.00	This command was modified to display OpenFlow hybrid port mode information.

# clear openflow

Clears flows from the flow table.

**Syntax** `clear openflow { flowid flow-id | all }`

**Parameters** `flowid flow-id`

`all` Clears the given flow ID that you want to delete from the flow table.

`all`

Deletes all flows from the flow table.

**Modes** User EXEC mode

Privileged EXEC mode

Global configuration mode

**Usage Guidelines** When an OpenFlow rule or all flows in the flow table need to be deleted you can use the **clear openflow** command with the **all** option. To delete a single OpenFlow rule based on a flow-id, use the **clear openflow** command with the **flowid *flow-id*** options.

**Examples** The following example clears the flow with an ID of 6.

```
device# clear openflow flowid 6
```

The following example clears all flows in the flow table.

```
device# clear openflow all
```

## History

---

Release	Command History
---------	-----------------

NI05.5.00c	This command was modified to delete a single flow on a specified flow-id or all flow deletion in the flow table.
------------	--

---

# show openflow groups

Displays the maximum number of actions in a bucket, the maximum number of buckets in a group and the maximum number of groups.

**Syntax** `show openflow groups group-id`

**Parameters** `groups group-id`  
Shows details of a specific OpenFlow group.

**Modes** User EXEC mode  
Privileged EXEC mode  
Global configuration mode

**Command Output** The `show openflow groups` command displays the following information:

Output field	Description
Group	Maximum number of group in a flow
Bucket	Number of bucket per group
Action	Number of action per bucket

```

device#show openflow groups
Max number of groups           : 512
Max number of buckets per group : 64
Max number of actions per bucket : 1

Max number of SELECT groups    : 64
Max number of buckets in SELECT group: 12
Starting Trunk ID for SELECT groups : 257
Group id 1

    Transaction id      4043243760
    Type                ALL
    Packet Count        0
    Byte Count          0
    Flow Count          0
    Number of buckets   2
bucket #1
    Weight              0
    Number of actions   1
        action 1: out port: 2/3

    bucket #2
    Weight              0
    Number of actions   1
        action 1: out port: 2/4

----
Total no. of entries printed: 1
    
```

History	Release version	Command history
	NI05.7.00	This command was introduced.

## show openflow queues

Displays the queues on the OpenFlow ports.

**Syntax** `show openflow queues ethernet Slot/port`

**Parameters** `ethernet Slot/port`

Gives information about a particular slot and port in an ethernet.

**Modes** User EXEC mode

Privileged EXEC mode

Global configuration mode

**Command Output** The `show openflow queues` command displays the following information:

Output field	Description
Queue	Number of queues
Rate	Minimum and maximum rate of the queue
Packet	Number of packet in the queue
Bytes	Number of bytes in the queue

**Examples** device#show openflow queues ethernet 2/1

```

Openflow Port    2/1
Queue 0
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 1
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 2
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 3
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 4
  Min Rate: 0           Max Rate: 0
  Tx Packets: 1918620
  Tx Bytes: 168838560
Openflow Port    2/1
Queue 5
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 6
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0
Openflow Port    2/1
Queue 7
  Min Rate: 0           Max Rate: 0
  Tx Packets: 0
  Tx Bytes: 0

```

**History**

Release version	Command history
NI05.7.00	This command was introduced.



# show openflow meters

Displays all the meters in a OpenFlow flow.

**Syntax** `show openflow meters meter-id`

**Parameters** `meters meter-id`

Shows details of a specific OpenFlow meter.

**Modes** User EXEC mode

Privileged EXEC mode

Global configuration mode

**Command Output** The `show openflow meters` command displays the following information:

Output field	Description
Meter-id	Meter number
Band	Number of bands in a meter
Band type	Band type ( supported type: Drop, DSCP_REMARK)
Rate	Rate of the band
Counter	Band specific counter

**Examples** The following example displays output with specific meter in MP.

```
device(config)# show openflow meters 2
Meter id: 2

Transaction id:      1438
Meter Flags:         KBPS BURST STATS
Flow Count:         0
Number of bands:    2
In packet count:    -NA-
In byte count:      0

Band Type:          DSCP-REMARK

Rate:                750000
Burst size:          1500          kb
Prec level:          1
In packet band count: -NA-
In byte band count:  0

Band Type:          DROP

Rate:                1000000
Burst size:          2000          kb
In packet band count: -NA-
In byte band count:  0

----
Total no. of entries printed: 1
```

The following example displays output with specific meter in LP.

```

device(config)# show openflow meters 1
Meter id: 1023

Meter Flags:
Number of bands:          KBPS BURST
                        2
RL Class Index:          33      33
In packet count:        -NA-
In byte count:           0

Band Type:      DROP

Rate:           3000          Adjusted rate:2996
Burst size:     1250          kb
In packet band count: -NA-
In byte band count: 0

Band Type:      DSCP-REMARK

Rate:           1700          Adjusted rate:1693
Burst size:     1250          kb
Prec level:     27
In packet band count: -NA-
In byte band count: 0

```

#### History

Release version	Command history
NI05.7.00	This command was introduced.

# clear statistics openflow

Clears OpenFlow statistics.

**Syntax** `clear statistics openflow { group | meter | controller }`

**Parameters**

**group** Clears statistics for all groups.

**meter** Clears statistics for all meters.

**controller** Clears statistics for all controllers.

**Modes** EXEC and Privileged EXEC mode  
Global configuration mode

**Usage Guidelines** This command can be entered in three configuration modes as shown in the examples below.

**Examples** The following example, entered in User EXEC mode, clears statistics for all groups in User EXEC mode.

```
device> clear statistics openflow group
```

The following example, entered in Privileged EXEC mode, clears statistics for all meters in Privileged EXEC mode.

```
device> enable
device# clear statistics openflow meter
```

The following examples, entered in global configuration mode, clears statistics for all controllers.

```
device# configure terminal
device(config) # clear statistics openflow controller
```

## History

Release	Command History
NI05.7.00	This command was introduced.

# openflow hello-reply disable

Allows the second Hello message (Hello-reply) to be disable on the OpenFlow Controller.

- Syntax** `openflow hello-reply disable`
- Command Default** This command needs to be run and saved when connecting to the OpenFlow Controller and any other controllers by default.
- Modes** EXEC and Privileged EXEC mode  
Global configuration mode
- Usage Guidelines** When the OpenFlow Controller receives the Hello message that the controller sent, it replies with another Hello message using the same transaction-ID as in the received Hello message.

```

Examples
device(config)# openflow ?
  controller      Configure controller
  default-behavior Default forwarding for no match packets
  enable          Enable/disable OpenFlow
  hello-reply     Configure HELLO Reply for HELLO originated from Controller

device(config)# openflow hello-reply ?
  disable        Disable HELLO Reply from the switch/router

device(config)# openflow hello-reply disable ?

device# show openflow
Administrative Status:      Enabled
SSL Status:                 Enabled
Source-Interface:          Not Configured
Source-Interface Status:   NA

Controller Type:           ofv130
HELLO Reply:               disabled
Number of Controllers:     2
.....

device# show running-config | i openflow
openflow enable ofv130
openflow hello-reply disable
    
```

History	Release version	Command history
	NI05.7.00	This command was introduced.