CONFIGURATION GUIDE

Extreme Network OS Software Defined Networking (SDN) Configuration Guide, 7.2.0

Supporting Network OS 7.2.0

53-1005150-02
15 June 2018

Contents

Preface......5

- Document conventions......5
  - Notes, cautions, and warnings......5
  - Text formatting conventions......5
  - Command syntax conventions......6
- Extreme resources......6
- Document feedback......6
- Contacting Extreme Technical Support......7

About this document......9

- Supported hardware and software......9
- Using the Network OS CLI......9
- What's new in this document......9

OpenFlow 1.3......11

- OpenFlow 1.3 Protocol overview......11
  - Flow table entries for NOS products......12
  - Supported OpenFlow instructions......13
  - Supported OpenFlow actions......14
  - OpenFlow counters......14
  - Supported OpenFlow messages......16
  - OpenFlow-hybrid switch forwarding-path operation......17
  - VRF support on OpenFlow ports......20
- Configuring OpenFlow......20
  - Prerequisites for configuring OpenFlow 20
  - Prerequisites for configuring an OpenFlow SSL connection 20
  - Configuring an OpenFlow controller......21
  - Creating an OpenFlow logical instance......21
  - Configuring an OpenFlow-enabled interface......22
  - Show commands for OpenFlow......23
  - Clear commands for OpenFlow......24
- Group table......24
  - Group messages......24
- Metering......25
  - Meter statistics......26
  - Meter messages......27

Preface

- Document conventions.......5
- Extreme resources.......6
- Document feedback.......6
- Contacting Extreme Technical Support.......7

Document conventions

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

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.

Text formatting conventions

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

Format	Description
bold text	Identifies command names. Identifies keywords and operands. Identifies the names of GUI elements. Identifies text to enter in the GUI.
italic text	Identifies emphasis. Identifies variables. Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	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
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	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.
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, *member*[*member*...].
\	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.

Extreme resources

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

White papers, data sheets, and the most recent versions of Extreme software and hardware manuals are available at www.extremenetworks.com. Product documentation for all supported releases is available to registered users at www.extremenetworks.com/support/documentation.

Document feedback

Quality is our first concern at Extreme, 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:

- Use our short online feedback form at http://www.extremenetworks.com/documentation-feedback-pdf/
- Email us at internalinfodev@extremenetworks.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.

Contacting Extreme Technical Support

As an Extreme customer, you can contact Extreme Technical Support using one of the following methods: 24x7 online or by telephone. OEM customers should contact their OEM/solution provider.

If you require assistance, contact Extreme Networks using one of the following methods:

- GTAC (Global Technical Assistance Center) for immediate support
  - Phone: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact.
  - Email: support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.
- GTAC Knowledge - Get on-demand and tested resolutions from the GTAC Knowledgebase, or create a help case if you need more guidance.
- The Hub - A forum for Extreme customers to connect with one another, get questions answered, share ideas and feedback, and get problems solved. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- Support Portal - Manage cases, downloads, service contracts, product licensing, and training and certifications.

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

About this document

- Supported hardware and software.....9
- Using the Network OS CLI.....9
- What's new in this document.....9

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Extreme Networks, Inc. for Network OS, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release of Network OS:

- ExtremeSwitching VDX 2746
- ExtremeSwitching VDX 6740
  - ExtremeSwitching VDX 6740-48
  - ExtremeSwitching VDX 6740-64
- ExtremeSwitching VDX 6740T
  - ExtremeSwitching VDX 6740T-48
  - ExtremeSwitching VDX 6740T-64
  - ExtremeSwitching VDX 6740T-1G
- ExtremeSwitching VDX 6940-36Q
- ExtremeSwitching VDX 6940-144S
- ExtremeSwitching VDX 8770
  - ExtremeSwitching VDX 8770-4
  - ExtremeSwitching VDX 8770-8

To obtain information about a Network OS version other than this release, refer to the documentation specific to that version.

Using the Network OS CLI

For complete instructions and support using the Network OS command line interface (CLI), refer to the Network OS Command Reference.

What's new in this document

This document supports the features introduced in Network OS 7.0.0 for Network OS 7.2.0:

For complete information, refer to the *Network OS Release Notes*.

OpenFlow 1.3

- OpenFlow 1.3 Protocol overview........11
- Configuring OpenFlow........20
- Group table........24
- Metering........25

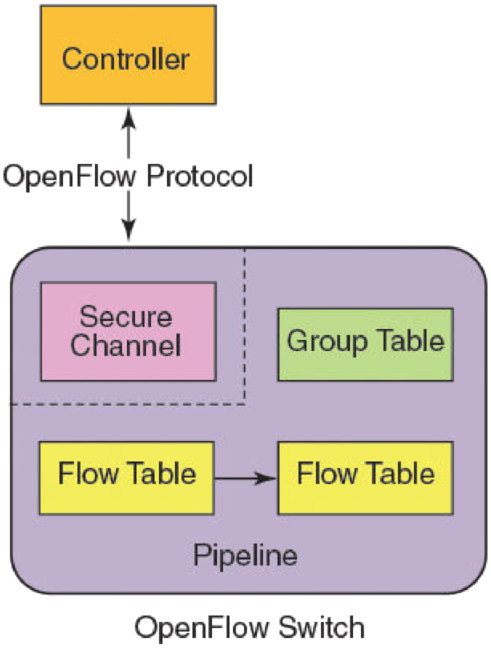
OpenFlow 1.3 Protocol overview

An OpenFlow-enabled switch 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 implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller.

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 1 OpenFlow 1.3 architecture

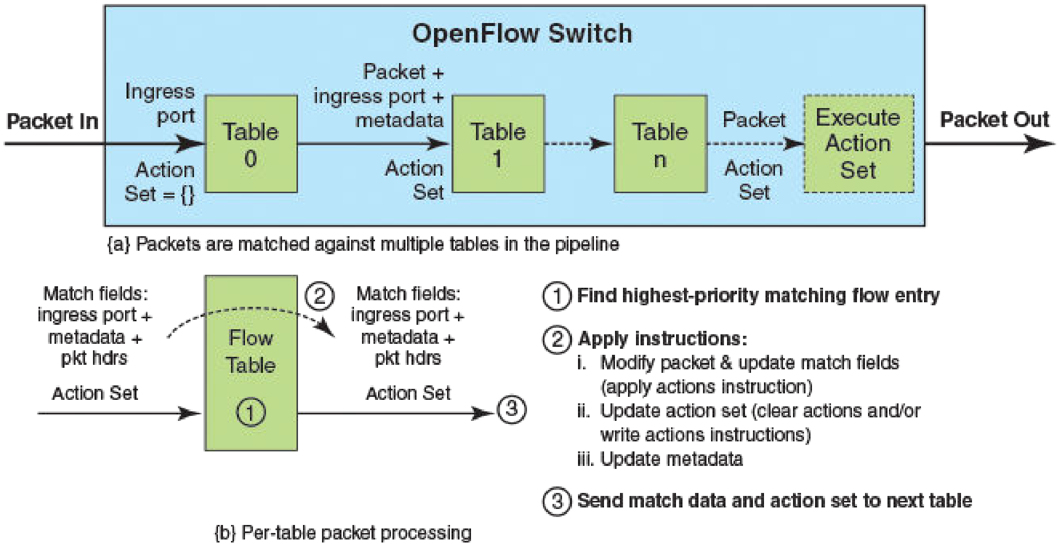


OpenFlow 1.3 defines three types of tables.

- Flow tables
- Group table
- Meter table

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

FIGURE 2 Pipeline processing


OpenFlow Switch
Packet In
Ingress port
Action Set = {}
Table 0
Packet + ingress port + metadata
Action Set
Table 1
Table n
Packet
Action Set
Execute Action Set
Packet Out
{a} Packets are matched against multiple tables in the pipeline
Match fields: ingress port + metadata + pkt hdrs
Action Set
Flow Table
1
2
Match fields: ingress port + metadata + pkt hdrs
Action Set
3
1 Find highest-priority matching flow entry
2 Apply instructions:
i. Modify packet & update match fields (apply actions instruction)
ii. Update action set (clear actions and/or write actions instructions)
iii. Update metadata
3 Send match data and action set to next table
{b} Per-table packet processing

Flow table entries for NOS products

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

TABLE 1 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 2 OpenFlow match fields

Match field	VDX 8770 series	VDX 6740 series	VDX 6940 series	Remarks
Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.	Yes	Yes	Yes	
Ethernet source MAC address	Yes	Yes	Yes	Layer 2 fields only
Ethernet destination MAC address	Yes	Yes	Yes	Layer 2 fields only
Ethernet type of the OpenFlow	Yes	Yes	Yes	
VLAN-ID	Yes	Yes	Yes	
VLAN priority (VLAN-PCP)	Yes	Yes	Yes	
IPv4 source address	Yes	Yes	Yes	Layer 3 fields only
IPv4 destination address	Yes	Yes	Yes	Layer 3 fields only
IPv4 protocol number	Yes	Yes	Yes	
IPv4 DSCP (IPv4 ToS bits)	Yes	Yes	Yes	Diff Serv Code Point (DSCP). Part of the IPv4 ToS field.
IPv4 ECN (IPv4 ToS bits)	No	No	Yes	
TCP/UDP source port	Yes	Yes	Yes	
TCP/UDP destination port	Yes	Yes	Yes	
ICMPv4 type	Yes	Yes	Yes	No partial matching support for ICMPv4_Type
ICMPv4 code	Yes	Yes	Yes	No partial matching support for ICMPv4_Code

Supported 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 devices.

> **NOTE**
> Only one flow table is supported on all platforms for OpenFlow instructions.

TABLE 3 Actions for flow table instruction

Actions	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Write-Action actions (Req)	Adds or overwrites specified actions to the action set.	Yes	Yes	Yes
Apply-Action actions	Applies the specified actions immediately.	Yes	Yes	Yes
Clear-Action actions	Clears all the actions in the action set.	No	No	No
Meter *meter-id*	Directs the packet to the specified meter.	Yes	Yes	Yes
Goto -Table *next-table-id* (Req)	Indicates the next table in pipeline processing.	No	No	No
Write-Metadata metadata/mask	Writes the metadata field from the mask.	No	No	No

Supported 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 the switch 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 4 Supported Actions for devices

Actions	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
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	Yes	Yes
Drop (Req)	No explicit drop action. Packet with empty action set should be dropped.	Yes	Yes	Yes
Group	Processes the packet through the specified group.	Yes	Yes	Yes
Set field	Modifies the values of the packet header based on the field type.	Yes (Layer 3: VLAN_VID, IPV4_DSCP; Layer 2: ETH_DST, VLAN_PCP)	Yes (Layer 3: VLAN_VID, IPV4_DSCP; Layer 2: ETH_DST, VLAN_PCP)	Yes (Layer 3: VLAN_VID, IPV4_DSCP; Layer 2: ETH_DST, VLAN_PCP)
Push-Tag/ Pop-Tag	Adds and removes tag (newly inserted tags are always the outermost tags).	Yes	Yes	Yes
Set-Queue	Set the queue ID for the packet.	Yes	Yes	Yes
Change TTL	Modify the TTL value.	Yes[1]	Yes[1]	Yes[1]

NOTE
1: Support for Decrement IP TTL action only.

OpenFlow counters

The devices record the number of received packets and bytes on a per-flow basis.

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

TABLE 5 Per-port OpenFlow counters supported on devices

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Received packets	Number of packets received on the port	Yes	Yes	Yes
Transmitted packets	Number of packets transmitted from the port	Yes	Yes	Yes
Received bytes	Number of bytes received on the port	Yes	Yes	Yes
Transmitted bytes	Number of bytes transmitted from the port	Yes	Yes	Yes
Receive drops	Number of received packets dropped on the port because the packets did not match any rules	No	Yes	Yes
Transmit drops	Number of transmit packets dropped on the egress port	No	Yes	Yes

TABLE 5 Per-port OpenFlow counters supported on devices (continued)

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Receive errors	Number of errors detected on the port on received packets	Yes	Yes	Yes
Transmit errors	Number of errors detected on the port on transmitted packets	Yes	Yes	Yes
Receive frame alignment errors	Number of frame alignment errors detected on packets received on the port	Yes	Yes	Yes
Receive overrun errors	Number of packets that caused overrun in the receive buffer on the port	Yes	Yes	Yes
Receive CRC errors	Number of packets received on the port that had CRC errors	Yes	Yes	Yes
Collisions	Number of collisions recorded on the port	No	No	No
Duration in seconds	Time	Yes	Yes	Yes

The following table lists the per-flow counters available:

TABLE 6 Per-flow OpenFlow counters supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Received packets	Number of packets received per flow	Yes	Yes	Yes
Received bytes	Number of bytes received per flow	Future support	No	No
Duration in seconds	Time	Yes	Yes	Yes

The following table lists the per-table counters available:

TABLE 7 Per-table OpenFlow counters supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Active entries	Number of active flow entries received per table	Yes	Yes	Yes
Packet look-ups	Number of packets received per table	No	No	No

NOTE
Active entry support is implemented in the software. Maximum flow entries may vary from device to device.

The following table lists the per-queue statistics available:

TABLE 8 Per-queue statistics supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Transmitted packets	Number of transmitted packets per queue	No	Yes	Yes

TABLE 8 Per-queue statistics supported (continued)

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Transmitted bytes	Number of transmitted bytes per queue	No	Yes	Yes
Transmit overrun errors	Number of packets that caused overrun in the transmit buffer on the port	No	No	No

The following table lists the per-group statistics available:

TABLE 9 Per-group statistics supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Packet count	Number of packets per group	Yes	Yes	Yes

The following table lists the per-meter statistics available:

TABLE 10 Per-meter statistics supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
Flow entry count	Number of transmitted bytes per queue	Yes	Yes	Yes
Input packet count	Number of packets per meter	No	No	No
Input byte count	Number of bytes per meter	Yes	Yes	Yes
Duration in seconds	Time	Yes	Yes	Yes

The following table lists the per-meter band statistics available:

TABLE 11 Per-meter band statistics supported

Counter	Description	VDX 8770 series	VDX 6740 series	VDX 6940 series
In-band packet count	Number of packets per meter band	No	No	No
In-band byte count	Number of bytes per meter band	Yes	Yes	Yes

Supported OpenFlow messages

The following OpenFlow messages are supported on the devices.

TABLE 12 OpenFlow messages

Message type	VDX 8770 series	VDX 6740 series	VDX 6940 series
OFPT_HELLO	Yes	Yes	Yes
OFPT_ERROR	Yes	Yes	Yes
OFPT_ECHO_REQUEST	Yes	Yes	Yes
OFPT_ECHO_REPLY	Yes	Yes	Yes
OFPT_EXPERIMENTER	No	No	No
OFPT_FEATURES_REQUEST	Yes	Yes	Yes

TABLE 12 OpenFlow messages (continued)

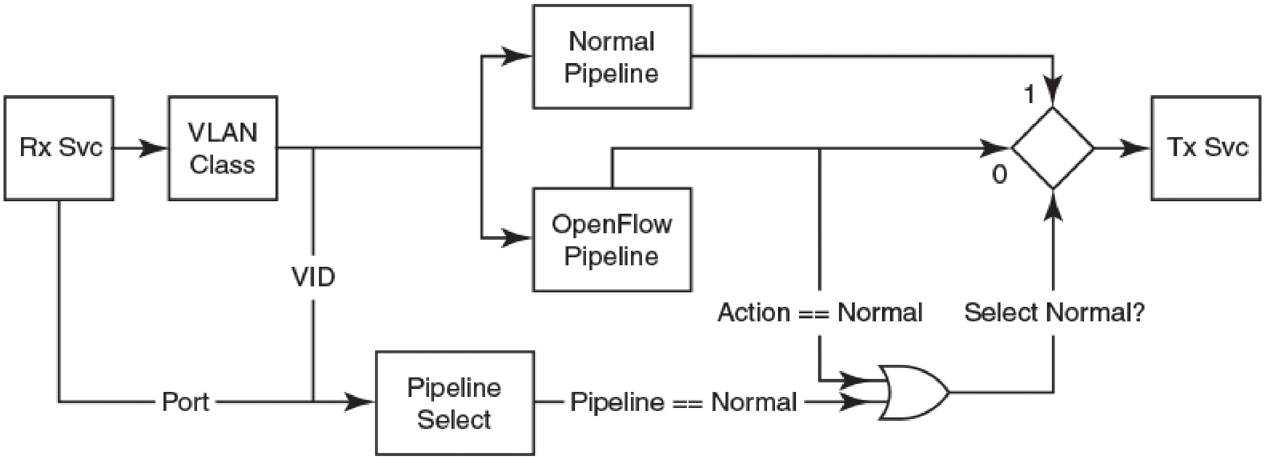
Message type	VDX 8770 series	VDX 6740 series	VDX 6940 series
OFPT_FEATURES_REPLY	Yes	Yes	Yes
OFPT_GET_CONFIG_REQUEST	No	No	No
OFPT_GET_CONFIG_REPLY	No	No	No
OFPT_SET_CONFIG	No	No	No
OFPT_PACKET_IN	Yes	Yes	Yes
OFPT_FLOW_REMOVED	Yes	Yes	Yes
OFPT_PORT_STATUS	Yes	Yes	Yes
OFPT_PACKET_OUT	Yes	Yes	Yes
OFPT_FLOW_MOD	Yes	Yes	Yes
OFPT_GROUP_MOD	Yes	Yes	Yes
OFPT_PORT_MOD	No	No	No
OFPT_TABLE_MOD	No	No	No
OFPT_MULTIPART_REQUEST	Yes	Yes	Yes
OFPT_MULTIPART_REPLY	Yes	Yes	Yes
OFPT_BARRIER_REQUEST	Yes	Yes	Yes
OFPT_BARRIER_REPLY	Yes	Yes	Yes
OFPT_QUEUE_GET_CONFIG_REQUEST	No	No	No
OFPT_QUEUE_GET_CONFIG_REPLY	No	No	No
OFPT_ROLL_REQUEST	Yes	Yes	Yes
OFPT_ROLL_REPLY	Yes	Yes	Yes
OFPT_GET_ASYNC_REQUEST	Yes	Yes	Yes
OFPT_GET_ASYNC_REPLY	Yes	Yes	Yes
OFPT_SET_ASYNC	Yes	Yes	Yes
OFPT_METER_MOD	Yes	Yes	Yes

OpenFlow-hybrid switch forwarding-path operation

OpenFlow supports the deployment of an OpenFlow switch in a network in two modes: OpenFlow-only switch mode and OpenFlow-hybrid switch mode. In OpenFlow-hybrid switch mode, the switch supports either OpenFlow traffic forwarding or normal traffic forwarding on individual ports.

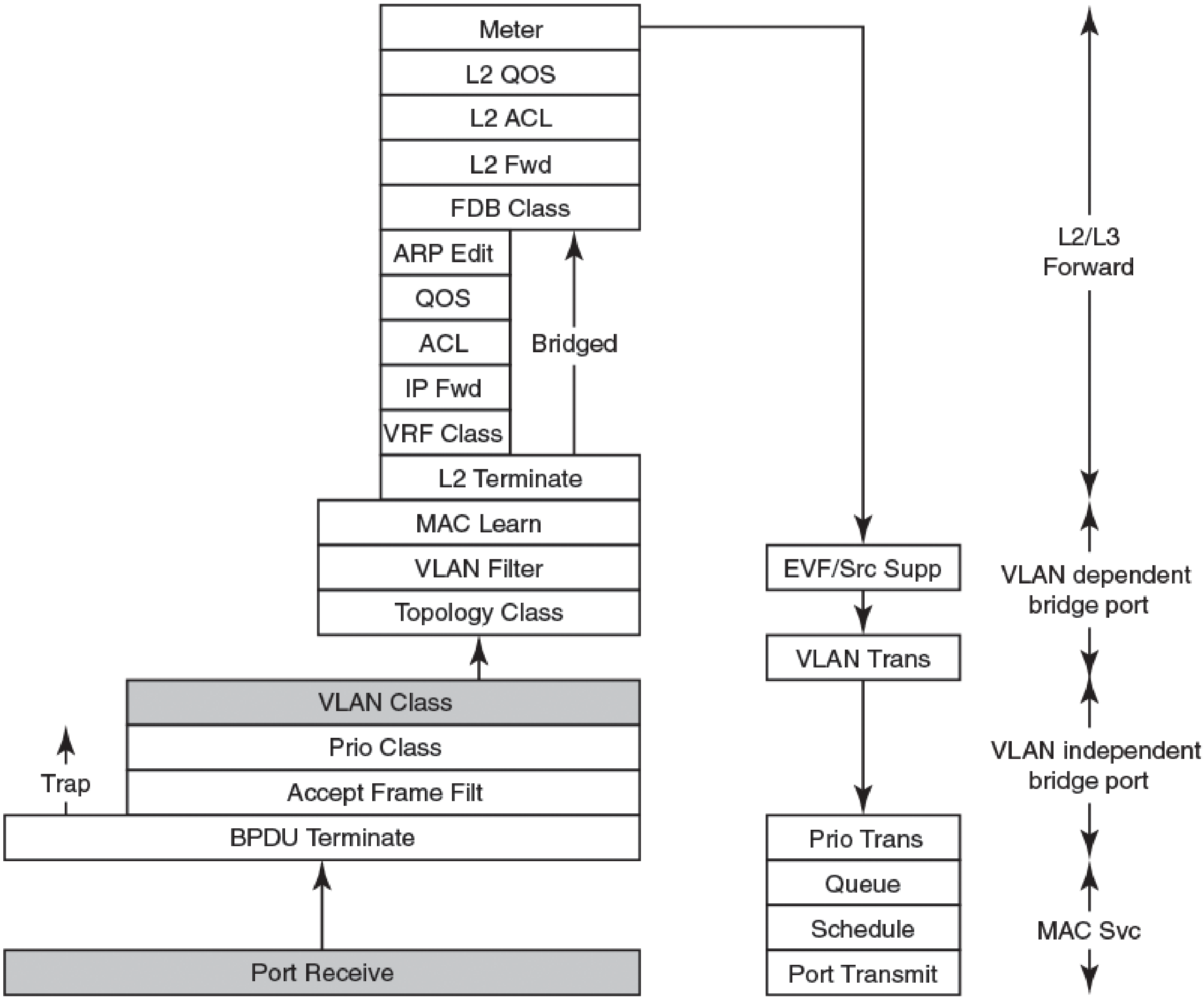
In OpenFlow switch mode, the OpenFlow controller exclusively controls all behavior of the forwarding path above the MAC services layer. In OpenFlow-hybrid switch mode (shown below), Normal forwarding path operates concurrently with the OpenFlow forwarding path. Frame forwarding decisions are confined to a single pipeline unless the OpenFlow pipeline (if selected) generates a special Normal output action, in which case the result of the Normal pipeline is selected.

FIGURE 3 OpenFlow hybrid switch mode pipeline



The pipeline selection mechanism is independent of OpenFlow protocol and are commonly supported by port-based selection and VLAN-based selection. The processing pipeline is shown below.

FIGURE 4 Processing pipeline model



The right column illustrates the conceptual separation between the MAC receive and transmit services, the bridge port functions (VLAN-dependent and VLAN-independent), and the forwarding function (Layer 2 and Layer 3). MAC services are contained in the Port Receive, Queue, Schedule, and Port Transmit layers.

Bridge Protocol Data Unit (BPDU) termination, acceptable frame filtering, priority classification, VLAN classification, and egress VLAN translation are performed by the VLAN-independent bridge port component. The acceptable frame filter, priority classifier, and VLAN classifier represent the scope of action of these components to all frames received from the MAC receive function except the trapped BPDUs (i.e. all data frames). The VLAN-dependent bridge port functions are comprised of Topology Classifier (VLAN scope of the topology control protocol such as MSTP), the VLAN filter, source MAC learning function, EVF, source suppression, and VLAN translation functions. The offset-arrow between the VLAN Classifier and the Topology Classifier, represents the per-VLAN de-multiplex process. The Topology Classifier, and its companion VLAN Filter, and MAC Learning function have reduced scope of action to the one or more VLANs comprising the Topology.

All components above the bridge port correspond to Layer 2 and Layer 3 forwarding function.

VRF support on OpenFlow ports

To reach a remote server or filter VRF based information, the system uses a particular VRF service. For reachability, you can configure the VRF for routing information to reach the application server.

VRF reachability indicates specification of the VRF for servicing requests from the clients and the clients specifying the VRF for reaching a source so that the management packets get serviced or routed in a specific VRF domain. This is done by the switch initiation or at the server.

The following are some examples of IP packet requests from the switch to use a particular VRF for the given service, SNMP notification, logging server, NSX controller, vCenter etc.

use-vrf gives the option to use a specified VRF. For an OpenFlow active controller, use this command:

```
device(config-openflow-controller-BVC) # ip address 10-24-82-10 use-vrf 1
```

Use this command for an OpenFlow passive controller:

```
device(config)# rbridge-id 1
device(config-rbridge-id-1)# openflow logical-instance 1
device(config-logical-instance-1)# passive use-vrf 1
```

Configuring OpenFlow

Configuring OpenFlow requires the following steps:

1. Configure the OpenFlow controller. Refer to Configuring an OpenFlow controller on page 21 for details.
2. Create an OpenFlow logical instance. Refer to Creating an OpenFlow logical instance on page 21 for details. That topic also describes how you must then also:
   a. Associate the controller to this logical instance.
   b. Activate the logical instance.
3. Enable OpenFlow on switch interfaces. Refer to Configuring an OpenFlow-enabled interface on page 22 for details.

Prerequisites for configuring OpenFlow

Before you can use the configuration commands described in this chapter, you must enable OpenFlow in the Rbridge hardware profile. To enable OpenFlow with a default hardware-profile use the following command syntax:

```
device# configure
device(config)# rbridge-id 14
device(config-rbridge-id-14)# hardware-profile route-table default openflow on
%INFO: The TCAM profile has been set to openflow.
%Warning: To activate the new profile config, please run 'reload system' on the target switch.
```

For additional hardware-profile options, refer the *Network OS Command Reference* for details.

Prerequisites for configuring an OpenFlow SSL connection

Normally, OpenFlow is used with an SSL (also known as TLS) connection, which provides a secure, encrypted communications channel. To set up an SSL OpenFlow connection, perform the following steps:

1. Set up a certificate authority (CA). CA set-up is described in industry-standard documentation.
2. Set up the switch certificate, CA certificate and private key as described in the *Network OS Security Guide*.

3. Add the switch certificate to the Java truststore of the OpenFlow controller. Refer to your controller user guide.

Configuring an OpenFlow controller

To configure an OpenFlow controller in active mode, you can specify the name for the controller. (This example assumes that you have already configured CA certificates and hardware-profiles.) Perform the following steps:

1. In privileged EXEC mode, issue the **configure** command to enter global configuration mode.
2. Enter **openflow-controller** followed by the name you want to assign to the controller.
3. Enter **ip address** *IPv4_address* **method** *connection-name* **port** *port-num* for the controller.
   - The method and port keywords are optional. Refer to the *Network OS Command Reference* for complete syntax.

   NOTE
   The supported connection methods are SSL and non-SSL. The default OpenFlow controller port on the VDX is 6633. The standard OpenFlow controller ports are typically 6633 or 6653.

The following example configures the controller in active mode.

```
device# configure
device(config)# openflow-controller BVC
device(config-openflow-controller-BVC)# ip address 10.24.82.10 method ssl port 6633
device(config-openflow-controller-BVC)#
```

Creating an OpenFlow logical instance

After you create an OpenFlow logical instance, you can associate an already-created OpenFlow controller with the logical instance for active controller use. You can also configure a number of options under the OpenFlow logical instance.

NOTE
Only one logical instance can be created, and only one controller can be associated with a logical instance.

To create an OpenFlow logical instance, perform the following steps:

1. In global configuration mode, issue the **rbridge-id** command to enter RBridge configuration mode .
2. Enter **openflow logical-instance** 1. (The logical instance must be 1.)

The following example configures the logical instance within RBridge ID 12.

```
device(config)# rbridge-id 12
device(config-rbridge-id-12)# openflow logical-instance 1
device(config-logical-instance-1)#
```

Once you create the logical instance, you are in OpenFlow logical-instance configuration mode, but the logical instance is not yet active.

Associating a controller with the OpenFlow logical instance

To associate a controller with the OpenFlow logical instance, enter the **controller** *name* command in OpenFlow logical-instance configuration mode. The controller name must already have been configured using the **Openflow-controller** command. For example, to associate an already-configured OpenFlow controller named BVC to the logical instance, you would enter the following command:

```
device(config-logical-instance-1)# controller BVC
```

Configuring a passive connection for an OpenFlow logical instance (optional)

To configure a passive connection on a logical instance, use the **passive no-ssl** command. This command includes an option to specify the port on the switch that the OpenFlow controller uses to establish an OpenFlow connection. For example, to configure a passive connection using port 6633, enter the following command:

```
device(config-logical-instance-1)# passive no-ssl port 6633
```

If you enter an IP address, only a controller from that IP address can connect to the device in passive mode. Enter **passive** *no-ssl* **ip address** *IPv4_address* command.

```
device(config-logical-instance-1)# passive no-ssl ip address 10.24.82.10
```

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.

Activating the logical instance for the OpenFlow controller

To activate a logical instance, use **activate** command, as shown below:

```
device(config-logical-instance-1)# activate
```

The **no** form of this command deletes the controller from the logical instance.

Configuring the default action (optional)

By default, the device drops packets that do not match any of the programmed flows. This is called a table miss. However, you can configure a device-level option to forward the packets to the controller instead of dropping them. When sending a packet to the controller, the device sends a copy of the packet to each of the configured controller connections.

To configure this behavior, enter **default-behavior send-to-controller** command:

```
device(config-logical-instance-1)# default-behavior send-to-controller
```

To enable the default behavior to drop the packet in case of table miss, enter the **default-behavior drop** command:

```
device(config)# default-behavior drop
```

Configuring an OpenFlow-enabled interface

To configure an OpenFlow-enabled interface, you must already have associated the OpenFlow controller with the logical instance.

Perform the following steps to configure an OpenFlow-enabled interface:

1. Enter the **interface** *interface_subtype* command with an associated RBridge ID and slot/port number from global configuration mode.
2. Enter **lldp disable** to disable the LLDP.
3. Enter **openflow logical-instance** *1* to associate the OpenFlow logical instance to the interface. Currently, the only valid value for *logical instance number* is 1.

   The logical instance must be activated before you can associate it to an interface.
4. To configure an OpenFlow enabled interface, enter the **openflow enable** command.

The following example configures the OpenFlow-enabled interface for the controller.

```
device(config)#interface TenGigabitEthernet 12/0/12
device(conf-if-te-12/0/12)# lldp disable
device(conf-if-te-12/0/12)# openflow logical-instance 1
device(conf-if-te-12/0/12)# openflow enable
```

Configuring OpenFlow-enabled interfaces for a specific Layer

By default, OpenFlow matches only occur on Layer 2 headers. To change this behavior, use the **openflow enable** *Layer3* command. To revert back to Layer 2 matching, use the **openflow enable** *Layer2* command.

To configure an OpenFlow enabled interface for Layer 2, enter **openflow enable** *Layer2* command. .

```
device(conf-if-te-12/0/12)# openflow enable Layer2
```

To configure an OpenFlow enabled interface for Layer 3, enter **openflow enable** *Layer3* command. .

```
device(conf-if-te-12/0/12)# openflow enable Layer3
```

Use the **no** form of the command to disable a OpenFlow-enabled interface.

Once an interface becomes OpenFlow enabled, a limited set of commands is allowed on that interface. Some QOS-related configuration commands are allowed, provided that they do not cause frame edits, because OpenFlow does not natively support any protocol constructs to configure QOS parameters. The following list shows the only commands that are allowed on an OpenFlow-enabled interface; refer to the *Network OS Command Reference* for details:

- mtu <no of bytes>
- speed <100/1000/1000-auto/10000/auto>
- shutdown/no shutdown
- qos flowcontrol tx <on/off> rx <on/off>
- cee default
- service-policy <in/out> <policy-name>
- qos random-detect traffic-class <0...7> red-profile-id <id>

Show commands for OpenFlow

Show commands for OpenFlow are included in the following table and described in detail in the *Network OS Command Reference*.

TABLE 13 Show commands for OpenFlow

Command	Description
show openflow	Shows all the OpenFlow configuration.
show openflow flow	Shows all the flows configured in the system flow table.
show openflow controller	Shows the status of all the controllers.
show openflow interface	Displays the ports with OpenFlow.
show openflow resources	OpenFlow usage of the resources.
show openflow queues	Shows the queue entries for the interface.
show openflow group	Shows all the groups in a flow.
show openflow meter	Shows all the meters in a flow.

Clear commands for OpenFlow

Clear commands for OpenFlow are included in the following table and described in detail in the *Network OS Command Reference*.

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/groups/meters configured in the system.

You can clear the flow statistics for all flows or for a specified flow.

TABLE 14 Clear commands for OpenFlow

Command	Description
clear openflow	Clears a single OpenFlow rule based on a Flow ID or deletes all flows/groups/meters configured in the system.
clear statistics openflow	Clears statistics for the controller, flow, all groups, or all meters.

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).

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

TABLE 15 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

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

Group messages

The following table describes the processing of group messages.

TABLE 16 Group messages

Group message type	Entry exists	Entry does not exist	Notes
Add (OFPGC_ADD)	Deny ADD. Return error message to controller	Add is processed	Subject to constraints below

TABLE 16 Group messages (continued)

Group message type	Entry exists	Entry does not exist	Notes
Mod (OFPGC_MODIFY)	Group parameters and action buckets are updated	Deny MOD. Return error message to controller	Need to confirm if Update/ Modify is implemented as delete followed by add in the driver.
Del (OFPGC_DELETE)	Group entry is deleted. Flows which are associated with this group are also removed.	No Error. Message ignored	If a DEL comes in that has flows associated with it, then delete those flows from the system.

Error conditions and messages

This table lists the error conditions and the error OPCODES sent to the controller. The error type is always OFPET_GROUP_MOD_FAILED.

TABLE 17 Group messages

Error condition	Opcode
Adding group, if group already exists	OFPGMFC_GROUP_EXISTS
When group allocation exceeds memory or system limit	OFPGMFC_OUT_OF_GROUPS
Group type is not supported	OFPGMFC_BAD_TYPE
In case of group modification or deletion, if group does not exist	OFPGMFC_UNKNOWN_GROUP
Number of buckets in a group (each device has different limit)	OFPGMFC_OUT_OF_BUCKETS
Number of actions in a bucket greater than 2	OFPGMFC_BAD_BUCKET
ALL: Not an output port action or set field VLAN_VID	OFPGMFC_BAD_BUCKET
SELECT/INDIRECT: Not an output port action or set field VLAN_VID or ETH_DA	OFPGMFC_BAD_BUCKET
MGID creation failed	OFPGMFC_EPERM
MGID update failed	OFPGMFC_EPERM

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.

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

TABLE 18 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 19 Meter bands supported on devices

Meter bands	Supported
DROP	Yes
DSCP_REMARK	No
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

The metering system supports the features in the following table.

TABLE 20 Metering capabilities supported for metering features

Feature	VDX 8770 series	VDX 6740 series	VDX 6940 series
Band types (bitmap)	DROP	DROP	DROP
Capabilities (bitmap)	KBPS, BURST, STATS	KBPS, BURST, STATS	KBPS, BURST, STATS
Maximum color value	2 (RED, GREEN)	2 (RED, GREEN)	2 (RED, 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)

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.

Following counters are associated with the meter band type.

- Band packet count
- Band byte count

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

Meter messages

The following table describes the processing of the meter messages.

TABLE 21 Meter messages

Meter message type	Entry exists	Entry does not exist	Notes
Add (OFPMC_ADD)	Deny ADD. Return error message to controller	Add is processed	Deny ADD due to lack of memory or internal error or due to restrictions (hardware or otherwise) limiting the number of bands. Return error message to controller as per standard.
Mod (OFPMC_MODIFY)	Meter parameters and bands are updated. New bands replace the existing bands.	Deny MOD. Return error message to controller	Meter MOD shall update meter parameters and new bands replace the existing bands. FWD layer shall apply these changes instantly or make before break mechanism.
Del (OFPMC_DELETE)	Meter entry and bands removed. Flows which are associated with this meter are also removed.	No Error. Message ignored	Only the meter identifier needs to be specified for the delete request.