

Aruba OpenFlow 1.3 Administrator Guide for ArubaOS-Switch 16.10



a Hewlett Packard
Enterprise company

Part Number: 5200-6771
Published: October 2019
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Optane®, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Chapter 1 About this guide	7
Applicable products	7
Switch prompts used in this guide	7
Chapter 2 Introduction	8
About OpenFlow	8
OpenFlow architecture	9
Virtualization mode	9
Aggregation mode	10
OpenFlow features and benefits	12
Administrative methods	13
Supported RFCs and standards	13
Interoperability	14
Chapter 3 Configuring OpenFlow	18
Configuration overview	18
Entering OpenFlow	18
Entering OpenFlow context	18
Entering OpenFlow instance context	18
Preparing for configuration	18
Enabling or disabling OpenFlow	19
Configuring OpenFlow instances	19
Configure table number for a flow table	20
Setting OpenFlow protocol version	21
OpenFlow instance mode	21
Configure OpenFlow instance members	22
Configuration commands	22
Flow location	25
Configuring listener ports	25
Configuring a controller	25
Associate OpenFlow instance with OpenFlow controller	26
Overriding the current exclusion list protocols	27
openflow instance	27
no openflow instance	28
show openflow instance	30
openflow instance override-protocol all	31
openflow instance override-protocol none	31
Securing the connection between an OpenFlow instance and the controller	32
Configuring auxiliary connections	33
Associating the auxiliary connection index with an OpenFlow instance	35
Configuring number of software flow tables per instance	35
OpenFlow instance connection interruption mode	35
Setting maximum backoff interval for an instance	36
Configuring IP Control Table Mode	36
Configure OpenFlow controller ports	36
Controller role change	36
Controller roles	37
Port modification	38

Port modification and OpenFlow versions.....	40
Configuring egress-only ports.....	41
Software and hardware rate limiting.....	42
Limiting the usage of hardware resources.....	42
Hardware statistics refresh interval.....	43
Custom table numbering.....	43
Implementation notes.....	45
Configuring VLANs.....	45
Configuring and verifying routing.....	45
Configuring physical and logical ports.....	46
Pipeline configuration commands.....	46
OpenFlow MAC group.....	46
OpenFlow pipeline.....	46
IP control table pipeline.....	47
Metadata.....	48
Command to configure source MAC group table on an instance.....	50
Destination MAC group table on an instance.....	50
Human readable data-path description.....	50
Overriding the default drop action of tables.....	51
Chapter 4 OpenFlow Meters.....	53
Meter statistics, scale, and limitations.....	53
Chapter 5 Group table.....	55
Group type ALL.....	55
Group type SELECT.....	55
Group Type INDIRECT.....	55
Group Type FAST FAILOVER.....	56
Group actions.....	56
Group statistics.....	56
Group scale.....	56
Group limitations.....	56
Chapter 6 OpenFlow per-flow rate limiting.....	58
QoS extensions.....	58
Create a limiter.....	58
Get limiter details.....	58
Support an OpenFlow flow with a limiter.....	58
Chapter 7 OpenFlow Multi-VLAN instance.....	59
Overview.....	59
Chapter 8 OpenFlow custom pipeline.....	60
Custom pipeline-model.....	60
Default Pipeline in custom mode.....	60
Pipeline modification process.....	65
Pipeline creation guidelines.....	71
Performance in custom pipeline model.....	72
Scale on custom pipeline Instance.....	73
Custom matches.....	73
Defining a custom match field.....	73

Programming a flow with match on custom match field.....	75
Facts.....	76
Chapter 9 OpenFlow Packet-Out.....	77
Determining the VLAN	77
Chapter 10 OpenFlow match on TCP flags and L4 port ranges.....	79
Matching on TCP flags and TCP/UDP port ranges.....	79
Experimenter match fields.....	79
Flow Mod Validations.....	80
TCP Flags/L4 port range matching in Custom Match Mode.....	81
Addition or modification of tag for single-tagged packets.....	81
Is/Is-not table.....	82
Restrictions.....	82
Error Messages.....	83
Chapter 11 Push and pop VLAN.....	84
Chapter 12 Show commands.....	85
Show OpenFlow information.....	85
Show global OpenFlow information.....	85
Show auxiliary connection information.....	86
Show OpenFlow controllers.....	86
Show OpenFlow flow table information.....	86
Show OpenFlow instance.....	87
show openflow instance test message-statistics.....	87
Viewing multiport-filter-limit.....	88
Show OpenFlow resources.....	88
Show OpenFlow instance group.....	90
Show OpenFlow instance flow table information.....	91
Show OpenFlow instance information.....	93
Show OpenFlow instance capabilities information.....	93
Viewing OpenFlow instance flow-table.....	94
Show OpenFlow instance flows.....	94
Show group information for a specific instance.....	97
Show per flow rate limiter information.....	99
Viewing message statistics for an instance.....	100
Show meter information for a specific instance.....	100
Viewing port statistics per instance.....	101
Chapter 13 Troubleshooting OpenFlow.....	102
Diagnostic Tools Overview and Usage.....	102
Debug OpenFlow.....	102
Error messages.....	103
Interoperability error messages.....	103
Controller error messages.....	104
VLAN error messages.....	105
Instance error messages.....	106
Troubleshooting scenarios and error messages.....	112
Reporting problems	113
OpenFlow Flow-Mod and Pipeline-Mod error message enhancements.....	113

Chapter 14 Websites.....	114
Chapter 15 Support and other resources.....	115
Accessing Hewlett Packard Enterprise Support.....	115
Accessing updates.....	115
Customer self repair.....	116
Remote support.....	116
Warranty information.....	116
Regulatory information.....	117
Documentation feedback.....	117
Flow classification on v1, v2, and v3 modules.....	118
Hardware match chart.....	118
OpenFlow 1.3 multi-table model and device modes.....	121
Flow table capabilities.....	122
Match/Set-Field.....	122
Instructions.....	124
Actions.....	125
Implementation notes.....	126
A hardware flow with an idle timeout of 10 seconds gets deleted even though packets match the flow within the idle timeout.....	126
Controller flows — flow in hardware and processing software.....	126
DUT matches and processes incoming untagged packets for VLAN id.....	127
Events that change the Operational Status of the OpenFlow instance.....	127
OpenFlow influence on CPU generated packets.....	128
OpenFlow 1.0 supports IP address masking.....	128
Virtualization mode versus Aggregation mode — VLAN tags in packet_in messages	129
Precedence level in meters.....	129
Support for miss_len field in ‘switch configuration’ messages.....	131
Once a controller deletes flows from Table 0, it has to re-add in order for traffic to flow through an OpenFlow switch	131
OpenFlow matching traffic destined for switch MAC address	132
OpenFlow custom pipeline implementation notes.....	132
Multi-VLAN implementation notes.....	133
Implementation notes for OpenFlow groups in hardware.....	133
Implementation notes.....	133
Configuring secure connection HPE VAN SDN controller.....	135
Service Insertion.....	137

This guide provides information on OpenFlow configuration and administration, OpenFlow command syntax descriptions, and troubleshooting actions.

Applicable products

This guide applies to these products:

Aruba 2920 Switch Series (J9726A, J9727A, J9728A, J9729A, J9836A)

Aruba 2930F Switch Series (JL253A, JL254A, JL255A, JL256A, JL258A, JL259A, JL260A, JL261A, JL262A, JL263A, JL264A, JL557A, JL558A, JL559A, JL692A, JL693A)

Aruba 2930M Switch Series (JL319A, JL320A, JL321A, JL322A, JL323A, JL324A)

Aruba 3810 Switch Series (JL071A, JL072A, JL073A, JL074A, JL075A, JL076A)

Aruba 5400R zl2 Switch Series (J9821A, J9822A, J9850A, J9851A, JL001A, JL002A, JL003A, JL095A)

Switch prompts used in this guide

Examples in this guide are representative and may not match your particular switch/environment. Examples use simplified prompts as follows:

Prompt	Explanation
switch#	# indicates manager context (authority).
switch>	> indicates operator context (authority).
switch(config)#	(config) indicates the config context.
switch(vlan-x)#	(vlan-x) indicates the vlan context of config, where x represents the VLAN ID. For example: switch(vlan-128)#.
switch(eth-x)#	(eth-x) indicates the interface context of config, where x represents the interface. For example: switch(eth-48)#.
switch-Stack#	Stack indicates that stacking is enabled.
switch-Stack(config)#	Stack(config) indicates the config context while stacking is enabled.
switch-Stack(stacking)#	Stack(stacking) indicates the stacking context of config while stacking is enabled.
switch-Stack(vlan-x)#	Stack(vlan-x) indicates the vlan context of config while stacking is enabled, where x represents the VLAN ID. For example: switch-Stack(vlan-128)#.
switch-Stack(eth-x/y)#	Stack(eth-x/y) indicates the interface context of config, in the form (eth-<member-in-stack>/<interface>). For example: switch(eth-1/48)#

This document provides the following:

- General steps for OpenFlow configuration and administration
- OpenFlow command syntax descriptions, including show commands
- OpenFlow troubleshooting commands and debug actions



NOTE: This document covers only the additional features and commands for administering OpenFlow on certain switches that use software version 15.10 or later.

About OpenFlow

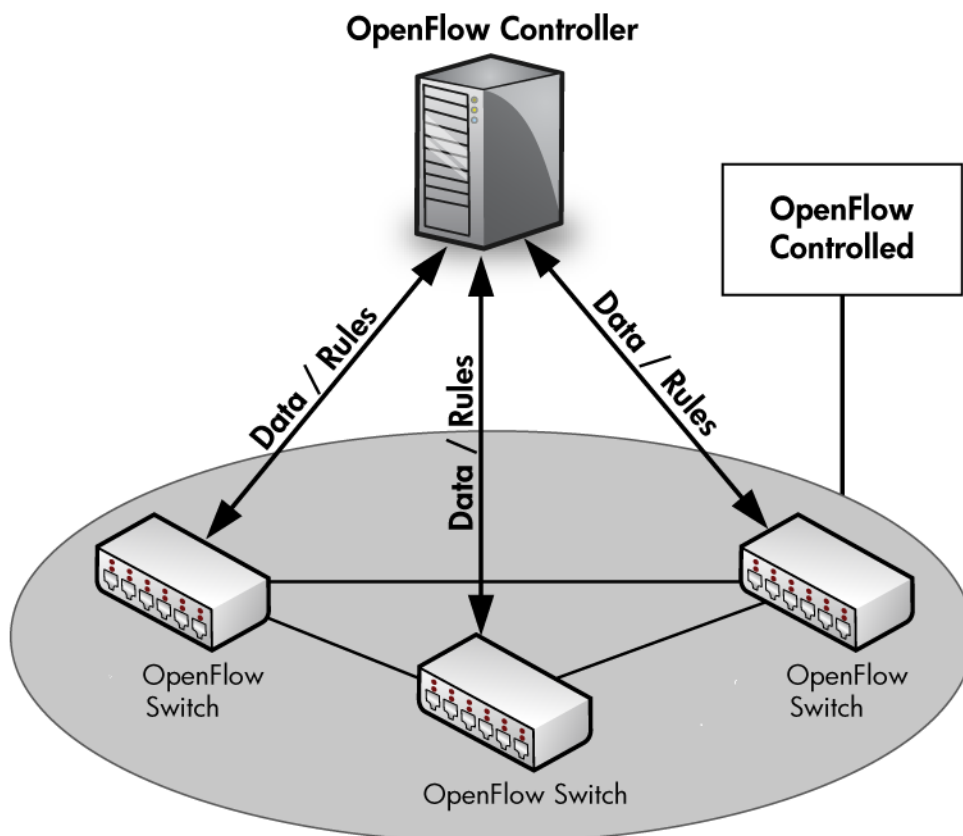
OpenFlow is a programmable open-standard network protocol that uses flexible matching rules to classify and manage network traffic into flows. OpenFlow defines a set of actions that network devices can take to manage these flows. An OpenFlow controller defines and communicates policies to specify traffic behavior on OpenFlow switches. OpenFlow separates the control plane (that decides how traffic must be forwarded) from the data plane (that implements how traffic is forwarded.)

OpenFlow is based on an Ethernet switch with internal flow-tables and a standardized interface to add and remove flow entries via an external controller.

OpenFlow is a software environment that allows for experimentation of networking protocols and traffic flows without interrupting the operation of a production network. OpenFlow traffic can be separated from the rest of the traffic on the network per VLAN so that OpenFlow does not impact non-OpenFlow traffic.

OpenFlow implementation on switches separates OpenFlow traffic and non-OpenFlow traffic with OpenFlow instances. Traffic within an OpenFlow instance does not influence or degrade non-OpenFlow traffic. OpenFlow configuration commands are applied per-instance.

Figure 1: *OpenFlow switches and controller*



HPE implementation complies with OpenFlow Switch Specification v1.0.0 (December 31, 2009.) With the K/KA.15.14, KB.15.15 (for the 5400R), KB.16.01 (for the 3810M), WB.15.14 and WC.16.02 releases, switches support OpenFlow Switch Specification v1.3.1 (September 2012). For implementation limitations with respect to the supported specifications, see **Supported RFCs and standards** on page 13.

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

OpenFlow architecture

OpenFlow can be configured to separate non-OpenFlow traffic from OpenFlow traffic. An OpenFlow instance can either be in the Virtualization or Aggregation Mode.

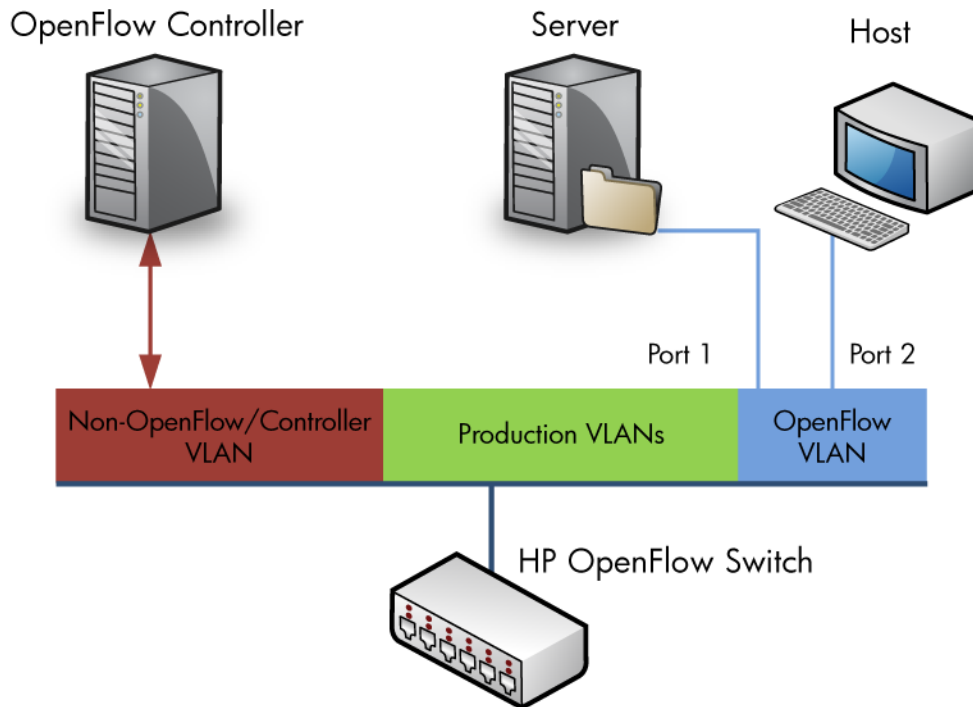
The communication channel between the OpenFlow Controller and the OpenFlow switch traverses the control-plane to communicate data and rules about how OpenFlow instances should operate. The OpenFlow instances/VLANs are the data-plane, which is where the OpenFlow rules are applied. Administrators must ensure that control-plane traffic does not traverse the data-plane, to avoid data-plane outages or other issues.

Virtualization mode

Virtualization mode allows non-OpenFlow VLANs and VLANs that belong to OpenFlow instances to be configured on the switch. Each OpenFlow instance is independent and has its own OpenFlow configuration

and OpenFlow controller connection. An OpenFlow instance in virtualization mode must have a VLAN associated as a member VLAN.

Figure 2: *Virtualization mode*



Aggregation mode

In Aggregation mode, all VLANs in the switch are part of an OpenFlow instance. The exception is the management VLAN and a VLAN that communicates to the controller. Similar to a lab environment the OpenFlow controller manages all the switching and routing for the switch.



NOTE: When Aggregation is configured, only OpenFlow traffic passes through the switch. Aggregation mode and virtualization mode are mutually exclusive (that is, virtualization mode cannot be configured when aggregation mode is configured and conversely).

Figure 3: Aggregation mode

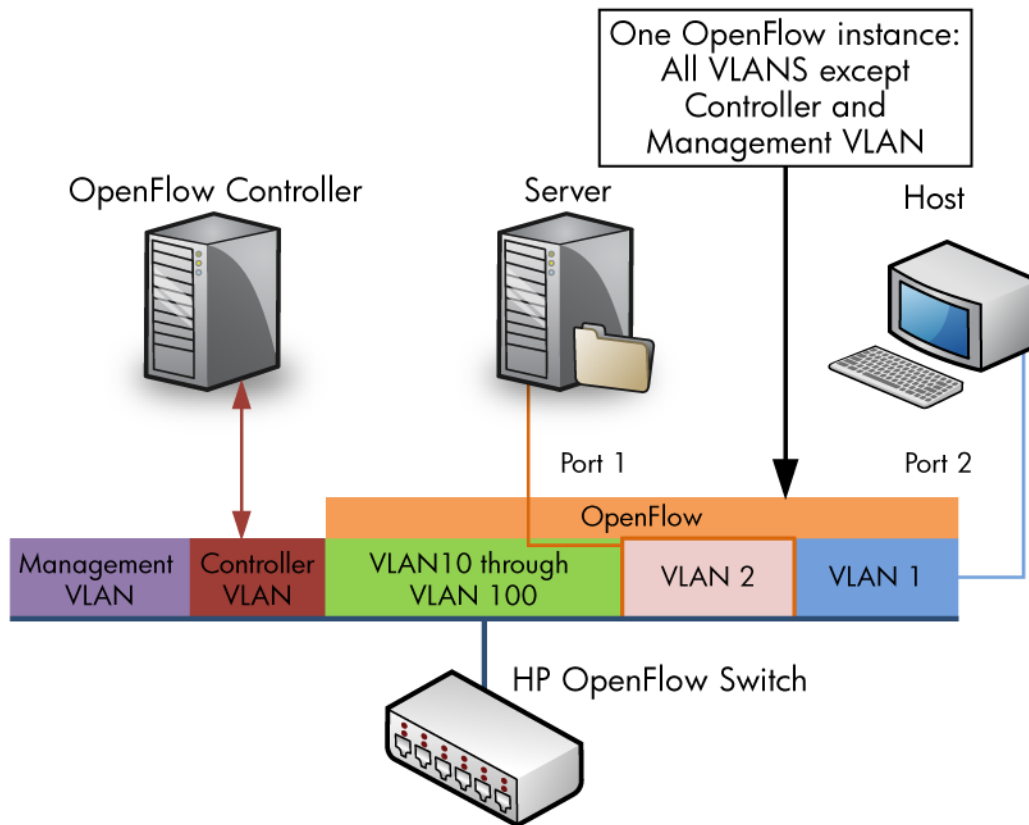
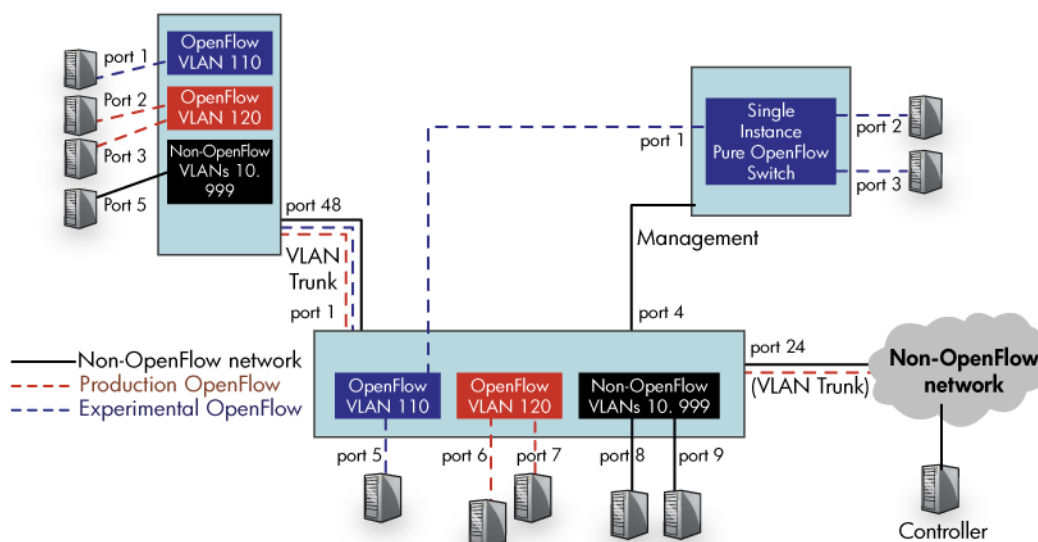


Figure 4: Example network with production non-OpenFlow, production OpenFlow, and experimental OpenFlow



OpenFlow features and benefits

- With the addition of OpenFlow Specification 1.3, the following features are supported:
 - Multiple Flow tables
 - Pipeline processing
 - Custom pipeline processing
 - Multi-VLAN instances
 - Groups in hardware
 - OpenFlow physical, logical, and reserved ports
 - Version negotiation
 - Group tables
 - Auxiliary connections
 - OpenFlow Extensible Match (OXM)
 - Multiple controllers
 - Support for IPv6 flows
- OpenFlow switch side configuration enables the user to:
 - Enable or disable OpenFlow
 - Create OpenFlow instances and configure controller connections
 - Display OpenFlow related configuration
 - Availability of Config support to retain OpenFlow configuration across a reboot
- OpenFlow supports high availability:
 - The OpenFlow flow table is preserved across Management Module failover
 - The OpenFlow configuration is synced from the AMM (Active Management Module) to the SMM (Standby Management Module).
- OpenFlow includes tools for limiting resources:
 - Support for limiting the percentage of policy engine and IP control table resources used by OpenFlow
 - Support for rate-limiting the amount of OpenFlow traffic sent to the controller
 - Support for rate-limiting the amount of OpenFlow traffic that gets forwarded by the policy engine rules programmed by OpenFlow
- OpenFlow modes of operation:
 - Support for hardware-only mode where only flows that can be programmed into hardware are accepted from the controller.
 - Support for active mode (default) where new flows are sent to the controller by the switch.
 - The switch normally handles support for passive mode where new flows no longer are sent to the controller.

IPv6 and OpenFlow

Directing IPv6 traffic using OpenFlow is supported beginning with OpenFlow Specification 1.3. For more information on configuring IPv6 on switches, see the *IPv6 Configuration Guide* for your switch.

Administrative methods

This document provides the HPE CLI commands for configuring and administering HPE OpenFlow switches.

The HPE VAN SDN controller has utilities for monitoring, administering, and troubleshooting OpenFlow switches. These utilities can show the current state of a switch that supports OpenFlow, including features, configuration, and table entries. Other controllers have similar utilities; for example, the OpenvSwitch controller distribution includes the utility `ovs-ofctl`.

Supported RFCs and standards

Switches support OpenFlow Switch Specification, version 1.0.0 (December 31, 2009) from the Open Networking Foundation, <https://www.opennetworking.org/> with some differences.

Unsupported features:

- TABLE action.
- The “enqueue” action per the specification.
- Handling of IP Fragments: OFPC_IP_REASM/OFPC_FRAG_REASM.
- The flow emergency cache implementation.
- OpenFlow 0.8.2 - SSL controller connection.
- OpenFlow 0.8.9 - Explicit handling of IP fragments: OFPC_IP_REASM.
- OpenFlow 0.8.9 - 802.1D Spanning Tree: OpenFlow does not allow full interaction with the switch Spanning Tree. OpenFlow is confined to the switch Spanning Tree.
- OpenFlow 0.9 - Emergency Flow Cache.
- OpenFlow 0.9 - Barrier Command.
- OpenFlow 1.0 - Slicing. The hardware cannot support the slicing specification. HPE provides its own QoS API instead.

Hardware acceleration limitations:

All flows cannot be executed in the switch hardware, because of hardware limitations. Flows not executed in hardware are processed in switch software, which is a slower path.

Switches support OpenFlow Switch Specification, version 1.3.1 (September 2012) from the Open Networking Foundation, <https://www.opennetworking.org/> with some differences.

Unsupported features:

- OFPP_TABLE action.
- Set-Queue action.
- Handling of IP Fragments: OFPC_IP_REASM/OFPC_FRAG_REASM.
 - Push-MPLS, Set MPLS TTL, Decrement MPLS TTL actions
 - Push-PBB action

- Copy TTL outwards, Copy TTL inwards actions
- Set queue action
- Some commands for port modification from a controller:
 - OFPPC_NO_STP
 - OFPPC_NO_RECV
 - OFPPC_NO_RECV_STP
 - OFPPC_NO_FWD



NOTE: When port modification commands are sent from the controller, an error message is returned to the controller: `OFPET_PORT_MOD_FAILED`.

Hardware differences between v1 and v2 Modules affect feature functionality, see **Flow classification on v1, v2, and v3 modules** on page 118 for details.

Interoperability

Table 1: *Switch features and interoperability with OpenFlow — by effect on feature or application*


Effect	Feature
Feature can override OpenFlow ¹	802.1X
	MAC Auth
	MAC Lockout
	MAC Lockdown
	Port Security
	Web Auth
Feature can override OpenFlow ²	ACLs – Port, VLAN, Router, IDM variants
	IDM
Feature can override OpenFlow ³	Rate Limiting
If OpenFlow is used, the feature can be configured.	Management VLAN
	 <p>NOTE: Management VLAN feature can be configured but it cannot be part of an OpenFlow instance.</p>

Table Continued

Effect	Feature
If OpenFlow is used, the feature cannot be configured. ⁴	Meshing
	Q-in-Q
	Remote Mirror Endpoint
	Transparent Mode
OpenFlow can override this feature ⁵	DHCP Snooping
	DHCPv4 client
	DHCPv4 relay
	DHCPv6 client
	DNS
	Ping
	SNTP
	Telnet client and server
	TFTP
	TimeP
	Traceroute
	BGP
OpenFlow can override this feature	DHCPv6 relay
	Dynamic ARP Protection
	Dynamic IP Lockdown
	IGMP Proxy
	IGMPv2
	IGMPv3
	MLDv1

Table Continued

Effect	Feature
	MLDv2 OSPFv2 OSPFv3 PIM-DM PIM-SM RIP Static Multicast Routes Static Routes Virus Throttling VRRP
OpenFlow does not affect this feature	Support existing L2, L3, security, HA, QoS functionalities
OpenFlow does not affect this feature ⁶	Distributed Trunking GVRP LACP Loop Protect sFlow UDLD
OpenFlow does not affect this feature ⁷	STP loop guard BPDU guard MSTP RSTP

Table Continued

Effect	Feature
	STP
	PVST

¹The authentication features still function in an OpenFlow instance and ports of an OpenFlow instance. The security features take a first look at the packet before sending the packets to OpenFlow.

²Any ACL entry that sets a drop bit in hardware (TCAM) always wins over the TCAM entry to copy OpenFlow traffic to the controller. Packets on an OpenFlow instance could then get dropped in hardware due to an ACL entry. An OpenFlow controller is never able to see those packets.

³Rate Limiting may be applied to limit OpenFlow traffic as well as other traffic. OpenFlow uses a form of rate-limiter to limit the OpenFlow traffic that gets to the CPU and to the controller.

⁴Enabling Meshing can break the distinction between OpenFlow VLANs and non-OpenFlow VLANs.

5

- The OpenFlow controller could set up a flow to match a protocol header and an action to drop the matching packets. This action could lead to the protocol packets never making it to the protocol handling code in the software data path, causing the protocol to break on the OpenFlow instance.
- The OpenFlow controller could set up a flow to match a protocol header and a NORMAL action in software for the matching packets. In such a case, OpenFlow removes the protocol packets in the software data path. OpenFlow reintroduces the protocol packets after examining the software flow table. Though this action may not break the protocol, it introduces an additional latency before the protocol running on the switch gets the protocol packets.

⁶Protocol packets are not sent through the OpenFlow software data path.

⁷Port up or down events are sent to the controller to keep the controller aware of available ports on the switch. OpenFlow cannot override STP, RSTP, or MSTP decisions.



NOTE: Following are the limitations when you enable OpenFlow and VxLAN together on the switch:

- When OpenFlow and VxLAN are enabled together on the same VLAN, the VxLAN tunnels are not advertised as an OpenFlow port to the controller. The Controller cannot program rules with match or output as VxLAN tunnels.
- When OpenFlow and VxLAN are enabled together on the switch but on different VLANs, all the packets tagged as unknown destination by the switch are not executed as per the OFPP_NORMAL action. Other OpenFlow actions such as output to a physical port or SI tap/ intercept tunnels work as expected.

Configuration overview

1. Enable OpenFlow.
2. Configure OpenFlow instances.
3. Configure OpenFlow instance members.
4. Set OpenFlow instance mode.
5. Set Flow location.
6. Configure software and hardware rate limiting.
7. Configure listener ports.
8. Configure controller IP and port.
9. Configure policy engine resources.

Entering OpenFlow

Entering OpenFlow context

You can use the `openflow` command options from configuration level by entering the word `openflow`, or from OpenFlow context level by typing the command option.

Syntax

```
openflow
```

Enters OpenFlow context

Entering OpenFlow instance context

You can use the `instance <instance-name>` command from configuration level by beginning the command with `openflow`, or from OpenFlow instance context level by typing the command option.

Syntax

```
openflow instance <instance-name>
```

Enters OpenFlow instance context

instance-name

OpenFlow instance name

Preparing for configuration

Plan your network including production and OpenFlow VLANs, OpenFlow instances, OpenFlow controller ports, listening ports, naming and numbering strategy.

Plan the number of VLANs configured for OpenFlow versus non-OpenFlow.

OpenFlow works on an instance only when OpenFlow is enabled on the instance as well as globally on the switch.



NOTE:

A maximum of 128 OpenFlow instances can be configured (16 on 2920 and 2930F switches). By default, a maximum of 2048 VLANs are supported, but if you change the MAX VLANs to 4096, OpenFlow supports all 4096 VLANs.

A maximum of 64000 OpenFlow rules can be added for 5400R/3810 (16000 for 2920 and 2930F).

A maximum of 128 controllers can be configured (16 for 2920 and 2930F).

A maximum software rate limit of 10000 pps can be configured (2000 for 2920 and 2930F).

Enabling or disabling OpenFlow

Enable or disable OpenFlow globally:

Syntax

```
openflow [enable | disable]
```

```
no openflow
```

enable

Enables OpenFlow globally.

disable

Disables OpenFlow globally.



NOTE: Using `no openflow` without any additional parameters deletes **all** OpenFlow configurations. A warning message displays to confirm this command.



NOTE: OpenFlow parameters can be changed only with OpenFlow disabled. Instance parameters cannot be changed when instance is enabled. To enable an instance, use the following command:

```
openflow instance <instance name> enable
```

Configuring OpenFlow instances



NOTE:

- Configuration changes are not allowed when instance is enabled. Disable the instance and make instance-specific configuration changes.
 - For a named instance to be enabled, a listen port or a controller, and a member VLAN must be added to the instance.
 - To enable an aggregate instance, a listen-port or a controller has to be added to the instance.
-

For more on Aggregation Mode, see [Aggregation mode](#) on page 10.

For more on Virtualization Mode, see [Virtualization mode](#) on page 9.

Syntax

```
openflow instance {instance-name | aggregate} [enable | disable]
```

```
no openflow instance {instance-name | aggregate} enable
```

The **no** form of the command deletes **all** OpenFlow configurations for the instance.

instance-name

Creates an OpenFlow instance.

Instance names can have a maximum length of 32 case-insensitive alphanumeric characters, numerals, and underscore.

aggregate

Creates an OpenFlow instance that includes all VLANs except the management VLAN and the OpenFlow controller VLANs. See **Aggregation mode** on page 10 for details on the use of this parameter.

enable

Enables the named OpenFlow instance or aggregate.

disable

Disables the named OpenFlow instance or aggregate.

Configure table number for a flow table

This command is used to configure a table number for a flow table. If the table number is not configured, a default value is given to the table. Table number configuration is supported only on standard-match pipeline-model.

If user does not configure table number for a given table, that table retains its default value.

If any table uses table number as 0, the default table 0 is not configured in the pipeline. In standard match pipeline-model, user can configure either policy table or the first software table as table 0. When the software table is configured as table 0, the OpenFlow pipeline has no H/W table at all.

Standard Match	
Flow Table	Table Number
Start	0
Policy Table	100
S/W Table 1	200
S/W Table 2	201
S/W Table 3	202
S/W Table 4	203

Syntax

```
openflow instance <instance-name> table-num  
no openflow instance <instance-name> table-num
```

```
switch(of-inst-t1)# table-num policy-table
```

Configure table number for flow tables.

policy-table

Specify the policy table number.

sw-table-#

Specify the software table 1 number.

<0-254>

Flow table number.

Flow policy table

```
switch(of-inst-t1)# table-num policy-table  
<0-254> Flow table number.
```

```
switch(of-inst-t1)# table-num sw-table-1  
<0-254> Flow table number.
```

```
switch(of-inst-t1)# table-num sw-table-2  
<0-254> Flow table number.
```

```
switch(of-inst-t1)# table-num sw-table-3  
<0-254> Flow table number.
```

```
switch(of-inst-t1)# table-num sw-table-4  
<0-254> Flow table number.
```

Setting OpenFlow protocol version

Syntax

```
openflow instance <instance-name> version {1.0|1.3[only] }
```

Default version: 1.0

OpenFlow protocol version supported by the instance.

This command lets you choose which version of OpenFlow the instance will use to negotiate with the controller. The command also allows for supported earlier versions of OpenFlow to be used in negotiation with the controller unless the option `only` is specified. See [Configure OpenFlow instance members](#) on page 22.

OpenFlow instance mode

OpenFlow can work either in *active* or *passive* mode.

Active mode

New packets of a flow that the switch is not aware of are sent to the OpenFlow controller.

Passive mode

There is one-way communication from the OpenFlow controller to the switch. Packets that do not match any flow in the flow table on the switch are not sent to the controller. The switch normally handles such packets of new flows.



NOTE: This option is applicable only for an OpenFlow version 1.0 instance.

This command sets operation mode for an OpenFlow instance.

Syntax

```
openflow instance <instance-name> mode {active | passive}
```

instance-name

Sets the mode for the named instance.

active

New flows are redirected to the controller for the instance.

passive

New flows are not sent to the controller for the instance.

Default: active

Configure OpenFlow instance members

- The same VLAN cannot be added as a member of multiple OpenFlow instances.
- The management VLAN cannot be added to an OpenFlow instance as a member VLAN.
- A Controller VLAN cannot be added to an OpenFlow instance as a member VLAN. For more information about multi-VLAN, see [Interoperability](#) on page 14.

Syntax

```
openflow instance <instance-name> member vlan <vlan-id>  
no openflow instance <instance-name> member vlan <vlan-id>
```

instance-name

Add a member to this OpenFlow instance.

vlan-id

Adds the VLAN to the named OpenFlow instance.

Configuration commands

Add member VLANs

Syntax

```
member vlan <vlan-id>
```

Used to add member VLANs to an instance.

Options

`no member vlan <vlan-id>` removes <vlan-id> from the multi-vlan instance. The other VLANs associated with the instance are still part of the instance. This command can take any of the following forms:

```
member vlan <vid1>
no member vlan <vid1>
```

```
member vlan vid1, vid2, vid3
no member vlan vid1, vid2, vid3
```

```
member vlan vid1 - vid3
no member vlan vid1 - vid3
```

```
member vlan vid1, vid3 - vid5, vid7
no member vlan vid1, vid3 - vid5, vid7
```

Add VLANs as members to OpenFlow instance

Syntax

```
openflow instance <instance-name> member vlan <VLAN-ID-LIST>
no openflow instance <instance-name> member vlan <VLAN-ID-LIST>
```

Add member VLANs to an OpenFlow instance. An aggregate instance is a special OpenFlow instance that includes all the VLANs on the switch except the management VLAN and controller VLAN.

<instance-name>

OpenFlow instance name.

<VLAN-ID-LIST>

List of VLANs that are added as member VLAN for an OpenFlow instance.

OpenFlow instance member VLAN

```
switch(config)# openflow instance <t1> member vlan 4,45-90,120
```

```
switch(config)# show running-config
Running configuration:
; J9850A Configuration Editor; Created on release #KB.15.17.0000x
; Ver #07:ff.f7.fc.7f.ff.3f.ef:a3
hostname "switch-name"
module A type j9992a
module F type j9986a
snmp-server community "public" unrestricted openflow
controller-id 1 ip 10.20.30.42 controller-interface vlan 2
instance "t1"
listen-port
member vlan 3,5-7,10,20,30
controller-id 1
```

```

version 1.3 only
flow-location hardware-only
pipeline-model custom
enable
exit
enable
exit

oobm
ip address dhcp-bootp
exit

vlan 1
name "DEFAULT_VLAN"
no untagged A1-A4,F23
untagged A5-A21,F1-F22,F24
ip address
exit

vlan 2
name "VLAN2"
untagged F23
ip address 10.20.30.40 255.255.255.0
exit

vlan 3
name "VLAN3"
untagged A1-A4
no ip address
exit

vlan 5
name "VLAN5"
no ip address
exit

vlan 6
name "VLAN6"
no ip address
exit

vlan 10
name "VLAN10"
no ip address
exit

vlan 30
name "VLAN30"
no ip address
exit
no allow-v2-modules

```

```

switch(config)# show openflow
OpenFlow                                     : Enabled
EgressOnly Ports Mode                       : Disabled

```

Instance Information

Instance Name	Oper. Status	No. of H/W Flows	No. of S/W Flows	OpenFlow Version
t1	Up	4	0	1.3 only

Flow location

This command sets the location of flows for an instance or the aggregate. In hardware-only mode, flows are programmed only in hardware. The flows are located in hardware and software by default.

Syntax

```
openflow instance <instance name> flow-location hardware-only
no openflow instance <instance name> flow-location hardware-only
```

instance-name

Sets flow location for the named instance.

hardware-only

Sets the location of flows to hardware-only.

Default: Software and hardware.



NOTE: If the flow cannot be added in hardware and the flow-location is set as hardware-only, an error is returned to the controller.

Configuring listener ports

Configures an OpenFlow port to listen for incoming connections from an OpenFlow controller.

Syntax

```
openflow instance <instance name> listen-port [tcp port] [oobm]
no openflow instance <instance name> listen-port [tcp port] [oobm]
```

instance-name

Sets the `listen-port` for the named instance.

tcp-port

Specify the port to listen on.

Default: Port number 6633

Range: Port number 1024 - 65534

oobm

Configure to listen through the out-of-band management (OOBM) port. This configuration is applicable only for switches that have a separate OOBM port.

Configuring a controller

Its IP address and connection port identifies a controller. Each OpenFlow instance can have up to three controllers. OpenFlow controllers can be added or deleted using this command.

Syntax

```
openflow controller-id <id> [ip < ip-address >] [port < tcp-port >] controller-interface {vlan < vlan-id > | oobm}
```

```
no openflow controller-id <id>
```

id

OpenFlow controller identification number.

If the controller is not in use by any OpenFlow instances, the `no` removes the identified controller.

Range: 1 - 128 (1 - 16 for 2920 and 2930F)

ip-address

OpenFlow controller IP address.

tcp-port

Optional: Specify the port through which to connect to a controller.

Default: 6633

Range: 1024 - 65534

controller-interface

The `no` form of the command with this parameter deletes the OpenFlow controller connection.

vlan-id

Connect to the OpenFlow controller through the identified VLAN.



NOTE: A VLAN that is a member of an OpenFlow instance cannot be added as an OpenFlow controller interface.

oobm

Connect to the OpenFlow controller through the OOBM interface. Applicable only for switches that have a separate out-of-band management (OOBM) port.

Associate OpenFlow instance with OpenFlow controller

Once the OpenFlow controller is set up, each instance must be associated to a controller.

Syntax

```
openflow instance <instance-name> controller-id <controller-ID>
no openflow instance <instance-name> controller-id <controller-ID>
```

Up to three controllers can be specified per OpenFlow instance.

The `no` removes the identified controllers.

instance-name

Sets controller for the named instance.

controller-ID

OpenFlow controller ID to be associated with the instance; up to three controllers per instance.

Example: Associating an OpenFlow instance with multiple controllers

To associate controllers 1, 5, and 100 to instance “test”, use the following commands:

```
switch (config)# openflow instance test controller-id 1
switch (config)# openflow instance test controller-id 5
switch (config)# openflow instance test controller-id 100
```



NOTE: When an OpenFlow controller is associated with an OpenFlow instance, it cannot be deleted.

Overriding the current exclusion list protocols

openflow instance

Syntax

```
openflow instance <INSTANCE-NAME> override-protocol  
[protocol-name]
```

Description

Control and view the state of OpenFlow exclusion list.

Options

802.1x

Allow OpenFlow to control 802.1x packets.

all

Allow OpenFlow to control all the protocols in this list.

bonjour

Allow OpenFlow to control BONJOUR packets.

dldp

Allow OpenFlow to control DLDP packets.

gvrp

Allow OpenFlow to control GVRP packets.

lACP

Allow OpenFlow to control LACP packets.

loop-protect

Allow OpenFlow to control LOOP-PROTECT packets.

mvrp

Allow OpenFlow to control MVRP packets.

none

Exclude all protocols in this list from being controlled.

pvst

Allow OpenFlow to control PVST packets.

smartlink

Allow OpenFlow to control Smartlink packets.

stp

Allow OpenFlow to control STP packets.

udld

Allow OpenFlow to control UDLD packets.

Usage

```
Switch (config)# openflow instance test override-protocol stp
```

Example

```
switch(config)# openflow instance test override-protocol stp

WARNING: Overriding the protocol can also potentially lead to control packets
         of the protocol to bypass any of the security policies like ACL(s).
Continue (y/n)? y

switch(config)# show running-config

Running configuration:

; JL256A Configuration Editor; Created on release #WC.16.03.0000x
; Ver #0e:3f.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:97
hostname "switch-name"
module 1 type jl256a
snmp-server community "public" unrestricted
openflow
  controller-id 1 ip 10.20.30.42 controller-interface vlan 2
  instance "test"
    member vlan 1
    controller-id 1
    version 1.3 only
    override-protocol stp
  exit
enable
exit
vlan 1
  name "DEFAULT_VLAN"
  no untagged 32
  untagged 1-31,33-52
  ip address dhcp-bootp
  exit
vlan 2
  name "VLAN2"
  untagged 32
  ip address 10.20.30.40 255.255.255.0
  exit
activate software-update disable

switch(config)# show openflow instance test override-protocol

Protocol      Override
-----
802.1x        No
bonjour       No
dldp          No
gvrp          No
lacp          No
loop-protect  No
mvrp          No
pvst          No
smartlink     No
stp           Yes
udld          No
```

no openflow instance

Syntax

```
no openflow instance <INSTANCE-NAME> override-protocol
```

Description

Add a protocol back to exclusion list.

Options

802.1x

Allow OpenFlow to control 802.1x packets.

all

Allow OpenFlow to control all the protocols in this list.

bonjour

Allow OpenFlow to control BONJOUR packets.

dldp

Allow OpenFlow to control DLDP packets.

gvrp

Allow OpenFlow to control GVRP packets.

lACP

Allow OpenFlow to control LACP packets.

loop-protect

Allow OpenFlow to control LOOP-PROTECT packets.

mvrp

Allow OpenFlow to control MVRP packets.

none

Exclude all protocols in this list from being controlled.

pvst

Allow OpenFlow to control PVST packets.

smartlink

Allow OpenFlow to control Smartlink packets.

stp

Allow OpenFlow to control STP packets.

udld

Allow OpenFlow to control UDLD packets.

Usage

```
protocol-name switch# no openflow instance <INSTANCE-NAME> override-protocol  
<PROTOCOL-NAME>
```

Example

```
switch(config)# openflow instance test override-protocol stp
```

```
WARNING: Overriding the protocol can also potentially lead to control packets  
         of the protocol to bypass any of the security policies like ACL(s).  
Continue (y/n)? y
```

```
switch(config)# show running-config
```

```
Running configuration:
```

```

; JL256A Configuration Editor; Created on release #WC.16.03.0000x
; Ver #0e:3f.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:97
hostname "switch-name"
module 1 type jl256a
snmp-server community "public" unrestricted
openflow
    controller-id 1 ip 10.20.30.42 controller-interface vlan 2
    instance "test"
        member vlan 1
        controller-id 1
        version 1.3 only
        override-protocol stp
    exit
enable
exit
vlan 1
    name "DEFAULT_VLAN"
    no untagged 32
    untagged 1-31,33-52
    ip address dhcp-bootp
    exit
vlan 2
    name "VLAN2"
    untagged 32
    ip address 10.20.30.40 255.255.255.0
    exit
activate software-update disable

switch(config)# show openflow instance test override-protocol

```

Protocol	Override
-----	-----
802.1x	No
bonjour	No
dldp	No
gvrp	No
lacp	No
loop-protect	No
mvrp	No
pvst	No
smartlink	No
stp	Yes
udld	No

show openflow instance

Syntax

```
show openflow instance <INSTANCE-NAME> override-protocol
```

Description

View the current state of the exclusion list.

Example

```
switch# show openflow instance <INSTANCE-NAME> override-protocol
```

```

switch# show openflow instance <INSTANCE-NAME> override-protocol
Protocol      Override
-----
802.1x        No

```

all	No
bonjour	No
dldp	No
gvrp	No
lacp	No
loop-protect	No
mvrp	No
pvst	No
smartlink	No
stp	No
traditional-pipeline	No
udld	No

openflow instance override-protocol all

Syntax

```
openflow instance <INSTANCE-NAME> override-protocol all
```

Description

Allow OpenFlow to control all the protocols.

Command context

Manager

Examples

```
switch#openflow instance test override-protocol all
WARNING: Overriding the protocol can also potentially lead to control packets
         of the protocol to bypass any of the security policies like ACL(s).
Continue (y/n)? y
switch(openflow) # show openflow instance test override-protocol
Protocol          Override
-----
802.1x            No
all               Yes
bonjour           No
dldp              No
gvrp              No
lacp              No
loop-protect      No
mvrp              No
pvst              No
smartlink         No
stp               No
traditional-pipeline No
```

openflow instance override-protocol none

Syntax

```
openflow instance <INSTANCE-NAME> override-protocol none
```

Description

Exclude all protocols from being controlled.

Command context

Manager

Examples

```
switch# openflow instance test override-protocol stp
Cannot add or remove individual protocols when "override-protocol all" is
configured.
Un-configure the same using "override protocol none" to proceed.
```

```
switch(openflow)# openflow instance test override-protocol none
switch(openflow)# openflow insatnce test override-protocol stp
switch(openflow)# show openflow instance test override-protocol
Protocol                Override
-----
802.1x                  No
all                     No
bonjour                 No
lldp                   No
gvrp                   No
lacp                   No
loop-protect           No
mvrp                   No
pvst                   No
smartlink              No
stp                    Yes
traditional-pipeline   No
udld                   No
```

Securing the connection between an OpenFlow instance and the controller

Syntax

```
controller-id <controller-ID> secure
no controller-id <controller-ID> secure
```

secure

Initiates a TLS connection with the controller (TLS version 1.0 or greater.)

controller-ID

OpenFlow controller ID to be associated with the instance.

This command:

- Secures the instance controller main connection. This option is available for OpenFlow version 1.0 as well as OpenFlow version 1.3.
- Supports CA signed certificates. For CA signed certificates, same ROOT certificate is used to sign both controller and switch certificate.
- Supports mutual authentication.

Example

```
switch(openflow)# show openflow instance test
Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
```



```

Member List           : VLAN 3
Pipeline Model        : Standard Match
Listen Port           : 6633
Operational Status    : Up
Operational Status Reason : NA
Datapath ID           : 000340a8f09e8600
Mode                  : Active
Flow Location         : Hardware and Software
No. of Hardware Flows : 6
No. of Software Flows : 4
Hardware Rate Limit   : 0 kbps
Software Rate Limit   : 100 pps
Conn. Interrupt Mode  : Fail-Secure
Maximum Backoff Interval : 60 seconds
Probe Interval        : 10 seconds
Hardware Table Miss Count : NA
No. of Software Flow Tables : 1
Egress Only Ports     : None
Table Model           : Policy Engine and Software
Source MAC Group Table : Disabled
Destination MAC Group Table : Disabled

```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Equal

Configuring auxiliary connections

Syntax

```
openflow # auxiliary-connection <index> port <port-number> type [tcp] [udp]
```

Creates an auxiliary connection with a unique index which is later associated with the instance controller main connection. Auxiliary connection uses the same source IP address and the datapath ID as the main connection. The main connection auxiliary ID is set to zero, while the auxiliary connection ID is set to 1. Only one auxiliary connection is supported per main connection and transport protocol options for auxiliary connections can be either TCP or UDP.

The packets supported on an auxiliary channel are:

- OFPT_HELLO
- OFPT_ERROR
- OFPT_ECHO_REQUEST/ REPLY
- OFPT_FEATURES_REQUEST/REPLY
- OFPT_PACKET_IN
- OFPT_PACKET_OUT

The main use of an auxiliary connection is for transactions related to message of type: OFPT_PACKET_IN/ OFPT_PACKET_OUT.

Options

index

Unique identifier for an auxiliary connection. Range: 1-128 (16 for 2920 and 2930F)

port-number

Protocol port on which the controller can be reached. Range: 1024-65534

tcp | udp

Type of transport protocol to be used.



NOTE: Auxiliary connections are terminated when the main connection goes down, or the user closes it, or when the OpenFlow instance/openflow is disabled, or OpenFlow is globally disabled. TLS is not supported for Auxiliary connections.

Example

```
switch(openflow)# auxiliary-connection 1 port 6633 type tcp
switch(openflow)# instance test
switch(of-inst-test)# controller-id 1 auxiliary-connection 1
switch(openflow)# show openflow instance test
Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model              : Standard Match
Listen Port                : 6633
Operational Status         : Up
Operational Status Reason  : NA
Datapath ID                : 000340a8f09e8600
Mode                       : Active
Flow Location              : Hardware and Software
No. of Hardware Flows      : 6
No. of Software Flows      : 4
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps
Conn. Interrupt Mode       : Fail-Secure
Maximum Backoff Interval   : 60 seconds
Probe Interval             : 10 seconds
Hardware Table Miss Count  : NA
No. of Software Flow Tables : 1
Egress Only Ports          : None
Table Model                : Policy Engine and Software
Source MAC Group Table     : Disabled
Destination MAC Group Table : Disabled
```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Equal

Controller Id	Auxiliary Conn. index	Auxiliary ID	Auxiliary Conn. Status	Auxiliary Conn. State	Type
1	1	1	Connected	Active	TCP

```
switch(openflow)# show running-config
```

```
openflow
controller-id 1 ip 10.20.30.42 controller-interface vlan 2
auxiliary-connection 1 port 6633 type tcp
instance "test"
listen-port
member vlan 3
controller-id 1 auxiliary-connection 1
version 1.3 only
```

```
enable
exit
enable
exit
```

Associating the auxiliary connection index with an OpenFlow instance

Syntax

```
openflow instance <instance-name> controller-id <controller-id> auxiliary-connection <index>
<controller-id>
```

OpenFlow controller ID to be associated with the instance. Range: 1-128 (1-16 for 2920 and 2930F)

index

Auxiliary connection index. Range: 1-128 (1-16 for 2920 and 2930F)



NOTE: Only one auxiliary connection is supported per main controller connection.

Configuring number of software flow tables per instance

Syntax

```
openflow instance<instance-name> software-flow-table <number-of-software-tables>
```

Configures the number of software flow tables required for an instance.

<number-of-software-tables>

Set the number of software tables.

Default: 1

Range: 1-4



NOTE:

This command is applicable only for an OpenFlow version 1.3 instance.

OpenFlow instance connection interruption mode

Use this mode to set behavior when an OpenFlow instance on the switch loses connection with the controller.

Syntax

```
openflow instance <instance-name> connection-interruption-mode {fail-secure | fail-standalone}
no openflow instance <instance-name> connection-interruption-mode {fail-secure | fail-standalone}
```

fail-secure

If this OpenFlow instance on the switch loses connection with all controllers, packets and messages intended for controllers are dropped. Flows continue to expire according to their time-outs.

Default: fail-secure

fail-standalone

If this OpenFlow instance on the switch loses connection with all controllers, legacy switching and routing functions handle packets of new flows. Existing flows of this OpenFlow instance are removed.

Setting maximum backoff interval for an instance

You can specify the maximum interval between two consecutive attempts to connect to a controller by an OpenFlow instance. The interval between two consecutive attempts increases exponentially until it reaches the specified value. All subsequent attempts use the specified value.

Syntax

```
openflow instance <instance-name> max-backoff-interval <seconds>
```

instance-name

OpenFlow instance name.

seconds

Maximum backoff interval time; Range: 1 — 3600

Default:

60

Configuring IP Control Table Mode

Deprecated syntax. See the command `no pipeline-model {standard-match | ip-control | custom}`.

Syntax

```
openflow no ip-control-table-mode
```

Includes IP control table in the OpenFlow packet processing pipeline.

Configure OpenFlow controller ports

An OpenFlow controller is configured globally under OpenFlow context and associated with an instance under instance context (see **Entering OpenFlow instance context** on page 18 for more information). OpenFlow controller traffic cannot be “in-band” or transit on a VLAN managed by OpenFlow. It must transit on a VLAN not managed by OpenFlow.

OpenFlow controller traffic and OpenFlow traffic can transit on the same port, as long as they use different VLANs.

The VLAN chosen for OpenFlow controller traffic depends entirely on the IP address of the controller, and no specific configuration is needed. Thus the switch must have a proper IP configuration, and the controller address must be part of a subnet that is not on an OpenFlow VLAN.

For information on how to either manually assign an IP address to the switch or set it up to perform DHCP queries, see the ‘Configuring IP Addressing’ chapter in the *Basic Operation Guide* for your switch.

Up to three OpenFlow controllers control each OpenFlow instance and each generates OpenFlow commands and data flows between an OpenFlow switch and the controller.

Controller role change

The following messages are related to controller role change:

OFPT_ROLE_REQUEST

Message from controller to the switch to change or query its role.

OFPT_ROLE_REPLY

Message sent in response to the OFPT_ROLE_REQUEST, returning the current role of the controller.

OFPT_SET_ASYNC

A controller, through this message can configure what asynchronous message notifications it wants to receive.

OFPT_GET_ASYNC

Controller uses this message to retrieve the asynchronous configuration set using the OFPT_SET_ASYNC message.



NOTE:

Whenever a connection is established between the switch and the controller, each controller starts in the role, OFPCR_ROLE_EQUAL. The controller can query and change its role if necessary.

Controller roles

By operating in different modes, controllers can synchronize handoffs in a scenario where multiple controllers are connected to the switch. Per the OpenFlow specification 1.3.1, a Controller can operate in one of the following roles:

- Equal
- Master
- Slave

Equal

Equal is the default role for a controller. The controller has full access to the switch and is equal to other controllers in the same role receiving all asynchronous messages from the switch (such as packet-in, flow-removed). Controller-to-switch commands are sent and modified within this role.

Slave

A Slave controller has read-only access to the switch. The controller cannot receive switch asynchronous messages except for Port-status messages. The controller is denied execution of the controller-to-switch commands: OFPT_PACKET_OUT, OFPT_FLOW_MOD, OFPT_GROUP_MOD, OFPT_PORT_MOD and OFPT_TABLE_MOD.

Master

The Master controller has full read-write access to the switch. Only one controller can be the Master at a given time. When the role of a controller is changed to Master, the switch changes all other controllers that it connects to, to a Slave role.

Syntax

To know the roles of controllers that a switch is connected to, use the following command:

```
show openflow instance <instance-name>
```

Example

```
switch(openflow)# show openflow instance test
```

```

Configured OF Version      : 1.3 only
Negotiated OF Version     : 1.3
Instance Name             : test
Data-path Description     : test
Administrator Status      : Enabled
Member List               : VLAN 3
Pipeline Model            : Standard Match
Listen Port               : 6633
Operational Status        : Up
Operational Status Reason : NA
Datapath ID               : 000340a8f09e8600
Mode                      : Active
Flow Location             : Hardware and Software
No. of Hardware Flows     : 6
No. of Software Flows     : 4
Hardware Rate Limit       : 0 kbps
Software Rate Limit       : 100 pps
Conn. Interrupt Mode      : Fail-Secure
Maximum Backoff Interval  : 60 seconds
Probe Interval            : 10 seconds
Hardware Table Miss Count : NA
No. of Software Flow Tables : 1
Egress Only Ports        : None
Table Model               : Policy Engine and Software
Source MAC Group Table    : Disabled
Destination MAC Group Table : Disabled

```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Slave
2	Connected	Active	No	Master

Port modification

Port modification is used to change the characteristics of a port in an OpenFlow instance on the switch via the controller. The controller sends an `OFP_PORT_MOD` message to the switch that can change the characteristics of a specific port.

The following command checks the state of the port configuration for all ports of an instance.

Syntax

```
show openflow instance <instance-name> port-statistics
```

Example

```

switch(of-inst-t1)# show openflow instance t1 port-statistics
Number of Ports :1
Port 1/1       : Up
Status
Admin. Status   : Enabled      Flood    : Enabled
Receive        : Enabled      Forward  : Enabled
Packet_in      : Disabled
Statistics
Collisions     : 0
Rx Packets     : 0             TxPackets : 47
Rx Bytes       : 0             TxBytes   : 10718
Rx Dropped     : 0             TxDropped : 0
Rx Errors      : 0             TxErrors  : 0
Frame Errors   : 0
CRC Errors     : 0
Overrun Errors : 0

```

Example: OpenFlow version 1.0

Wireshark Capture of a sample Port-Mod message for a 1.0 instance

```
OpenFlow Protocol
Header
  Version: 0x01
  Type: Port Mod (CSM) (15)
  Length: 32
  Transaction ID: 4
Port Modification
  Port #: 5
  MAC Address: HewlettP_02:2c:bb (84:34:97:02:2c:bb)
Port ConfigFlags
  ....0 = Port is administratively down: No (0)
  ...0. = Disable 802.1D spanning tree on port: No (0)
  ...0.. = Drop non-802.1D packets received on port: No (0)
  ...0... = Drop received 802.1D STP packets: No (0)
  ...1.... = Do not include this port when flooding: Yes (1)
  ...0..... = Drop packets forwarded to port: No (0)
  ...0..... = Do not send packet-in msgs for port: No (0)
Port Config Mask
  ....0 = Port is administratively down: No (0)
  ...0. = Disable 802.1D spanning tree on port: No (0)
  ...0.. = Drop non-802.1D packets received on port: No (0)
  ...0... = Drop received 802.1D STP packets: No (0)
  ...1.... = Do not include this port when flooding: Yes (1)
  ...0..... = Drop packets forwarded to port: No (0)
  ...0..... = Do not send packet-in msgs for port: No (0)
Port Advertise Flags
  ....0 = 10 Mb half-duplex rate support: No (0)
  ...0. = 10 Mb full-duplex rate support: No (0)
  ...0.. = 100 Mb half-duplex rate support: No (0)
  ...0... = 100 Mb full-duplex rate support: No (0)
  ...0.... = 1 Gb half-duplex rate support: No (0)
  ...0..... = 1 Gb full-duplex rate support: No (0)
  ...0..... = 10 Gb full-duplex rate support: No (0)
  ...0..... = Copper medium support: No (0)
  ...0..... = Fiber medium support: No (0)
  ...0..... = Auto-negotiation support: No (0)
  ...0..... = Pause support: No (0)
  ...0..... = Asymmetric pause support: No (0)
Pad: 0
Pad: 0
Pad: 0
Pad: 0
```

Example: OpenFlow version 1.3

Wireshark Capture of a sample Port-Mod message for a 1.3 instance

```
OpenFlow Protocol
Header
  Version: 0x04
  Type: Port Mod (CSM) (16)
  Length: 40
  Transaction ID: 4043243760
Port Modification
  Port #: 2
  Pad: 0
  Pad: 0
  Pad: 0
  Pad: 0
  MAC Address: HewlettP_02:2c:be (84:34:97:02:2c:be)
  Pad: 0
  Pad: 0
  Port ConfigFlags
  ....0 = Port is administratively down: No (0)
  ...0. = Disable 802.1D spanning tree on port: No (0)
```

```

.....0.. = Drop non-802.1D packets received on port: No (0)
.....0... = Drop received 802.1D STP packets: No (0)
.....0.... = Do not include this port when flooding: No (0)
.....0..... = Drop packets forwarded to port: No (0)
.....1.... = Do not send packet-in msgs for port: Yes (1)
  Port Config Mask
.....0 = Port is administratively down: No (0)
.....0. = Disable 802.1D spanning tree on port: No (0)
.....0.. = Drop non-802.1D packets received on port: No (0)
.....0... = Drop received 802.1D STP packets: No (0)
.....0.... = Do not include this port when flooding: No (0)
.....0..... = Drop packets forwarded to port: No (0)
.....1.... = Do not send packet-in msgs for port: Yes (1)
  Port Advertise Flags
.....0 = 10 Mb half-duplex rate support: No (0)
.....0. = 10 Mb full-duplex rate support: No (0)
.....0.. = 100 Mb half-duplex rate support: No (0)
.....0... = 100 Mb full-duplex rate support: No (0)
.....0.... = 1 Gb half-duplex rate support: No (0)
.....0..... = 1 Gb full-duplex rate support: No (0)
.....0..... = 10 Gb full-duplex rate support: No (0)
.....0..... = Copper medium support: No (0)
.....0..... = Fiber medium support: No (0)
.....0..... = Auto-negotiation support: No (0)
.....0..... = Pause support: No (0)
.....0..... = Asymmetric pause support: No (0)
  Pad: 0
  Pad: 0
  Pad: 0
  Pad: 0

```

Example

Send a Port-Mod command to the switch using dpctl, a controller utility.

```

root@openflow-ubuntu-10:/home/openflow# dpctl ltcp:10.20.30.50:6633 port-desc
... {no="6", hw_addr="00:1b:3f:cf:76:fa", name="A6", config="0x0", state="0x1",
curr="0x0", adv="0x0", supp="0x0", peer="0x0", curr_spd="100000000kbps",
max_spd="100000000kbps"} ...

root@openflow-ubuntu-10:/home/openflow# dcctl tcp:10.20.30.50:6633 port-mod
port=6,addr=00:1b:3f:cf:76:fa,conf=0x40,mask=0x40

SENDING:
port_mod{port="6", hwaddr="00:1b:3f:cf:76:fa", config="0x00000040",
mask="0x40", adv="0x0"}
OK

root@openflow-ubuntu-10:/home/openflow# dcctl tcp:10.20.30.50:6633 port-desc
... {no="6", hw_addr="00:1b:3f:cf:76:fa", name="A6", config="0x40",
state="0x1", curr="0x0", adv="0x0", supp="0x0", peer="0x0",
curr_spd="100000000kbps", max_spd="100000000kbps"} ...

```

Port modification and OpenFlow versions

An OpenFlow 1.0 instance on the switch supports OFP_PORT_CONFIG, OFPPC_NO_FLOOD and OFPPC_NO_PACKET_IN.

An OpenFlow v1.3 instance on the switch supports OFP_NO_FLOOD and OFPPC_NO_PACKET_IN.

**NOTE:**

If a port is not exclusive to the OpenFlow Member VLAN, a Port Modification message from the controller results in an error returned to the controller.

Configuring egress-only ports

This CLI command enables or disables support for advertising egress-only ports to the controller. Ports that are members of non-OpenFlow VLANs are egress-only ports. A controller can add a flow with an egress-only port as an output port to allow traffic to be forwarded from an OpenFlow VLAN to a non-OpenFlow VLAN. All instance member ports and egress-only ports are exposed as instance ports to the controller.

Syntax

```
openflow egress-only-ports
```

egress-only-ports

Enable or disable support for advertising egress-only ports to the controller.

Ports that are members of non-OpenFlow VLANs are egress-only ports. A controller can add a flow with an egress-only port as an output port to enable traffic to be forwarded from an OpenFlow VLAN to a non-OpenFlow VLAN.



NOTE: Egress-only ports cannot be used as an “in-port” in any flow by a controller. If this usage is attempted, the flow addition fails and an error message is returned to the controller.

Example

```
switch(openflow)# show openflow instance test
```

```
Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model              : Standard Match
Listen Port                : 6633
Operational Status         : Up
Operational Status Reason  : NA
Datapath ID                : 000340a8f09e8600
Mode                       : Active
Flow Location               : Hardware and Software
No. of Hardware Flows      : 6
No. of Software Flows      : 4
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps
Conn. Interrupt Mode       : Fail-Secure
Maximum Backoff Interval   : 60 seconds
Probe Interval             : 10 seconds
Hardware Table Miss Count  : NA
No. of Software Flow Tables : 1
Egress Only Ports          : None
Table Model                : Policy Engine and Software
Source MAC Group Table     : Disabled
Destination MAC Group Table : Disabled
```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Slave
2	Connected	Active	No	Master



NOTE: When the Egress-Only Ports option is enabled for OpenFlow on the switch, the Port-Mod message for an egress-only port results in an error.

Software and hardware rate limiting

You can specify resource limits used by an OpenFlow instance. Each OpenFlow instance has independent rate-limiters that can be set separately.

Syntax

```
openflow instance <instance-name> limit {hardware-rate | software-rate}
```

instance-name

OpenFlow instance name.

hardware-rate

Limit the bandwidth that an OpenFlow instance can utilize. The hardware-rate is in kbps.

Range: 0 - 10,000,000

Default: 0

software-rate

Configure the OpenFlow instance packet rate limit. The software-rate is in pps.

Limits the number of packets per second per module that this instance can send to the software path.

Range: 0 - 10000

Default: 100



NOTE:

Increasing the software rate limit increases CPU consumption and may impact system performance.

If the software rate limit is specified beyond 1000 pps, the following warning message is displayed: Increasing the software rate limit would increase CPU consumption and may impact the system performance.

Limiting the usage of hardware resources

Syntax

```
openflow limit {policy-engine-usage | ip-ctrl-table-usage | multiport-filter-usage} <max-percentage>
```

policy-engine-usage

Maximum percentage of policy engine resources used by OpenFlow.

ip-ctrl-table-usage

Maximum percentage of IP control table resources used by OpenFlow.

multiport-filter-usage

Maximum percentage of the multiport-filter resources used by OpenFlow.

You can limit the OpenFlow usage of policy engine resources, ip control table, and multiport filters so that other functions that use the same resources are not impacted severely.

The limit can be set only when OpenFlow is disabled globally.

percentage

Specifying 0% allocates no resources for OpenFlow.

By default, the OpenFlow policy engine resource usage is set at 50% to avoid oversubscribing resources and impacting performance. The policy engine resource can use Access Control Lists, Quality of Service, Identity Driven Management, Virus Throttling, Mirroring, Policy Based Routing, and other features in addition to OpenFlow.



NOTE: The maximum percentage is not a guaranteed percentage but a maximum allowed limit.

To increase the number of flows beyond the default 50% setting, use the above OpenFlow limit policy-engine-usage command. If all policy engine resources are in use, OpenFlow rules are no longer added in hardware and the switch denies attempts to configure ACLs with the CLI. See “Monitoring Resources” in the latest *Management and Configuration Guide* for your switch.

Default: 50%; Range: 0 - 100%



NOTE: Resource usage can be set only when OpenFlow is disabled.

Example

```
switch(openflow)# limit multiport-filter-usage
<0-100>                Enter a number.
```

```
switch(openflow)# show openflow multiport-filter-limit
```

Total Multiport Filters: 1039

Features	Filters Allocated	Filters Used	Filters Free
-----	-----	-----	-----
OpenFlow	519	1	518

Hardware statistics refresh interval

Syntax

```
openflow hardware-statistics refresh-interval policy-engine-table <seconds>
```

<seconds>

Refresh interval

Default: 20

Range: 0 - 3600



NOTE:

With value of 0, the hardware is no longer polled to update the statistics.

Custom table numbering

Enables a user to configure custom table numbers for tables in an OpenFlow pipeline. If the table number is not configured, a default value is given to the table.

Syntax

```
openflow instance <instance-name> table-num policy-table <table-number>
no openflow instance <instance-name> table-num policy-table <table-number>
```

Configure custom table-number.

policy-table

Specify the policy table number.

table-num

Specify the software table 1 number.

<0-251>

Flow table number.

Configure table number for flow tables.

```
switch(of-inst-t1)# table-num policy-table
```

Changing the table-number for a policy-table

```
switch(of-inst-t1)# table-num policy-table 1
switch(of-inst-t1)# enable
switch(of-inst-t1)# show openflow instance t1 flow-table
```

OpenFlow Instance Flow Table Information

Table ID	Table Name	Flow Count	Available Free Flow Count	Miss Count	Goto Table
0	Start	1	NA	0	1
1	Policy Table	1	NA	0	200
200	SW Table	1	1	0	*

* Denotes that the pipeline could end here.

Policy table as the first table in the instance.

```
switch(of-inst-t1)# table-num policy-table 0
switch(of-inst-t1)# enable
switch(of-inst-t1)# show openflow instance t1 flow-table
```

OpenFlow Instance Flow Table Information

Table ID	Table Name	Flow Count	Available Free Flow Count	Miss Count	Goto Table
0	Policy Table	1	NA	0	200
200	SW Table	1	1	NA	*

* Denotes that the pipeline could end here.

Software table as the first table in the instance.

```
switch(of-inst-t1)# disable
switch(of-inst-t1)# no table-num policy-table
switch(of-inst-t1)# table-num sw-table-1 0
switch(of-inst-t1)# enable
switch(of-inst-t1)# show openflow instance t1 flow-table
```

```
OpenFlow Instance Flow Table Information
Table
ID      Table Name      Flow      Available Free Miss
Count   Count             Count      Count      Count      Goto Table
-----
0       SW Table           1         1          NA          *
* Denotes that the pipeline could end here.
```

Implementation notes

- Custom table numbering command is supported only on Standard match mode instances.
- This feature is not supported on OpenFlow v1.0 instances.
- Different OpenFlow instances can have different table numbers for a given table.
- A `no` version of the command restores the table number to its default value.
- Only `policy-table` and `SW-table-1` can be configured as the first table in standard match mode pipeline.
- The default table numbers in a standard match mode instance are

Standard Match Mode	
Flow Table	Table Number
Start	0
Policy Table	100
Sw Table 1	200
Sw Table 2	201
Sw Table 3	202
Sw Table 4	203

Configuring VLANs

For information on configuring and verifying VLANs, see the *Advanced Traffic Management Guide* for your switch.

Configuring and verifying routing

For information on configuring and verifying routing, see the *Multicast and Routing Guide* for your switch.

Configuring physical and logical ports

For information on configuring and verifying ports, see the ‘Port Status and Configuration’ chapter in the *Management and Configuration Guide* for your switch.

Pipeline configuration commands

Syntax

```
pipeline-model {standard-match | ip-control | custom}  
no pipeline-model {standard-match | ip-control | custom}
```

Configure an OpenFlow instance pipeline model. By default, the instance is configured to use the standard-match pipeline model.

standard-match

Configure a standard-match pipeline model. The standard-match pipeline model enables an instance to advertise its policy-engine and software tables.

ip-control

Configure an ip-control pipeline model. The ip-control pipeline model enables an OpenFlow version 1.3 instance to advertise its IP control table along with policy-engine and software tables.

custom

Configure a custom pipeline model. The custom pipeline model enables an OpenFlow controller to customize an OpenFlow pipeline model for OpenFlow version 1.3 instance.

An instance without “pipeline-model” configured defaults to standard-match.

OpenFlow MAC group

MAC Group Tables allow the controller to apply the same policy to a set of users, differentiated by the source MAC address, using a single rule in the policy table. This application is achieved by exposing the MAC CAM as a separate table to controllers and allows controllers to create MAC groups.

OpenFlow pipeline

MAC group tables can be enabled for an OpenFlow instance in “standard match” mode or “ip control” mode. By default, both **source mac group** table and **destination mac group** tables are disabled. These tables can

be enabled in the pipeline by using the commands `src-mac-grp-table` or `dest-mac-grp-table`. The following figures depict the various combinations of OpenFlow pipelines that an instance can have.

Figure 5: *Standard match mode default pipeline*



Figure 6: *Standard match SRC enabled pipeline*

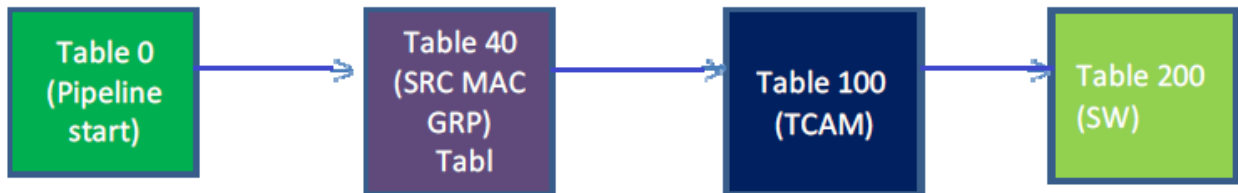
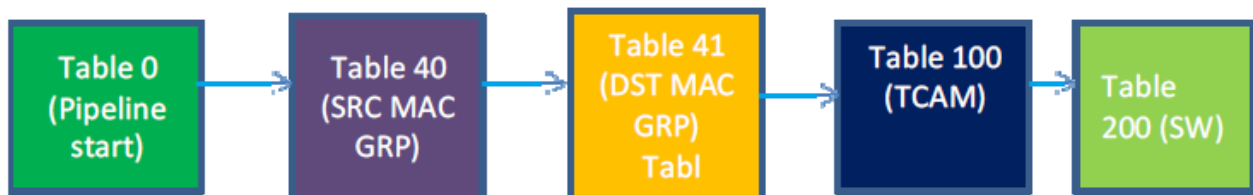


Figure 7: *Standard match DST enabled pipeline*



Figure 8: *Standard match SRC and DST enabled pipeline*



IP control table pipeline

A single consolidated policy table combines “IP Match Policy Table”, “IP Miss Policy Table” and “Non-IP Policy table”. “Policy table” in IP control table mode allows the controller to match on L3 match, L3 miss and L3

ignore. OpenFlow controller must program rules into policy table with metadata to achieve functionality. Table 3 illustrates the values of metadata and metadata mask that must be programmed.

Figure 9: IP control destination MAC enabled pipeline



Figure 10: IP control source MAC enabled pipeline

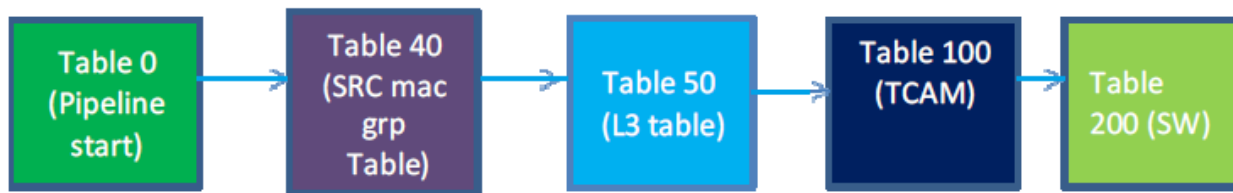


Figure 11: IP control table default



Metadata

The controller must program rules with match on metadata to achieve functionality.

The following table illustrates the values of metadata and metadata mask with which these functionalities can be achieved.

Table 2: Metadata

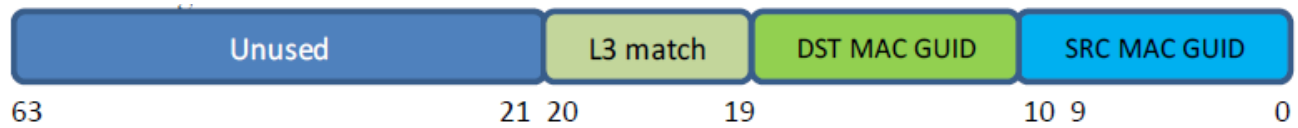
	Metadata	Metadata_mask
L3 match	1	1
L3 miss	0	1
L3 ignore	0	0
Non-IP		

OpenFlow supports metadata, which is a maskable register value that is used to carry information from one table to the next.

OpenFlow metadata

For example, `OFPXMT_OFB_METADATA` is part of `oxm_ofb_match_fields` and is a 64-bit field. It is used to pass information between lookups across multiple tables. This value can be arbitrarily masked. Out of 64 bits, the current release is using 21 bits and the following table illustrates the interpretation of these bits.

Figure 12: *OpenFlow metadata*



- SRC MAC GRP
 - 10 bits, which identifies the Source Mac Group ID
- DST MAC GRP
 - 10 bits, which identifies the destination Mac Group ID
- L3 match
 - 1 bit, which is used to identify L3 match/miss for IP control table mode.

The following table shows the values of `Metadata_match` and `metadata_write` bits advertised for a different table.

Table 3: *Bits advertised*

Table #	Metadata_match	Metadata_write
0	0	0
40	0	0x3FF (Bits 0 to 19) (Bits 0 to 9 are used to write the src GUID for MAC)
41	0	0xFFC00 (Bits 10 to 19) (Bits 0 to 9 are used to write the dst GUID for MAC)
50	0	0x100000 (Bit 20)
100	0x01FFFF (Bits 0 to 20)	0
200	0	0

**NOTE:**

- Rules on the policy table must match on metadata, which in turn is mapped to a MAC-group (source/destination MAC).
- The policy table cannot match on a source MAC and destination MAC address separately when MAC group tables are part of the pipeline.
- In L3 tables, rules can be added with instructions `WRITE_METADATA/MASK` and `GOTO`.
- On the policy table, the L3 match, the metadata, metadata_mask specified in the rule interprets L3.

Command to configure source MAC group table on an instance

Enable the source MAC group table in the OpenFlow pipeline.

Syntax

```
openflow instance <instance-name> src-mac-grp-table  
no openflow instance <instance-name> src-mac-grp-table
```

Enable the source MAC group table in the OpenFlow pipeline.



NOTE: The `src-mac-grp-table` is disabled by default.

Destination MAC group table on an instance

Enable the destination MAC group table in the OpenFlow pipeline

Syntax

```
openflow instance <instance-name> dest-mac-grp-table  
no openflow instance <instance-name> dest-mac-grp-table
```

<instance-name>

OpenFlow instance name



NOTE: The `dest-mac-grp-table` is disabled by default.

Human readable data-path description

The OFPMP_DESC multipart response message allows an OpenFlow switch to specify a human readable text of size 255 ASCII characters.

Syntax

```
Openflow instance <instance-name> datapath-desc <datapath text>
```

Set a name to the data-path description field in the OFPMP_DESC message. By default, the data path description is set to the instance name.

Set a name to the data-path description field

```
switch(of-inst-test)# datapath-desc dpid1
switch(of-inst-test)# show openflow instance test
```

```
Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : dpid1
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model             : Standard Match
Listen Port                : 6633
Operational Status        : Up
Operational Status Reason  : NA
Datapath ID                : 000340a8f09e8600
Mode                       : Active
Flow Location              : Hardware and Software
No. of Hardware Flows      : 6
No. of Software Flows     : 4
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps
Conn. Interrupt Mode       : Fail-Secure
Maximum Backoff Interval   : 60 seconds
Probe Interval             : 10 seconds
Hardware Table Miss Count  : NA
No. of Software Flow Tables : 1
Egress Only Ports          : None
Table Model                : Policy Engine and Software
Source MAC Group Table     : Disabled
Destination MAC Group Table : Disabled
```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Equal

Overriding the default drop action of tables.

As of 16.02, when an OpenFlow instance is enabled with version configured to 1.3 or 1.3 only, all the traffic directed to the OpenFlow member VLAN is dropped. Overriding the default drop action of tables when installed by a switch enabled in an OpenFlow instance version 1.3 (or 1.3 only) will normalize and minimize traffic loss. An enable command overrides the action.

Operating notes

- This feature is supported on all three generations of ASIC — v1, v2, and v3.
- This feature is supported on all three pipeline-models — Standard-Match, IP-Control, and Custom.
- This behavior is applicable even when the controller is not connected to the instance except for `output-controller + version 1.3` where packets are forwarded normally until the OpenFlow handshake completes.
- The `default-miss` actions are reinstalled after the controller connects and the instance exits the `fail-standalone` mode.

Restrictions

- This feature is available only for instances configured with version 1.3 or 1.3 only.
- The switch will not chain the pipeline and update the miss rule on the last table (**except for the implicit rules in Standard-Match and IP-Control pipeline-models**). This default-miss-action will be applied on individual tables.
 - In standard-match, there is a miss rule on Table 0 which will still have the action as GOTO 100 irrespective of the miss action set with this command.
 - In ip-control, there are two miss rules, one on Table 0 that has action as GOTO 50 and the other on Table 50 that has action as GOTO 100. These two rules again will not be affected by the setting of this command.

Example

The actions are applicable only on tables that support it.

For example, for Table 0 of the standard-match pipeline, the default action will still be `Goto-100`.

The OpenFlow meters are available on all products mentioned in the **Applicable products** section.

Meter types

Supports the following meter types:

- `OFPMF_KBPS`
- `OFPMF_PKTPS`

Meter bands and rates

Supports the following bands in meters:

- `OFPMBT_DROP`
- `OFPMBT_DSCP_REMARK` (not supported for v1 platforms)

Considerations for a meter band

- A minimum of one band is required.
- A single OpenFlow meter can have either one or two bands.
- Does not support two bands of type `OFPMBT_DROP`.
- The rate associated with the `OFPMBT_DROP` band must be higher than the rate associated with the `OFPMBT_DSCP_REMARK` band.
- The precision level associated with an `OFPMBT_DSCP_REMARK` band must be less than or equal to seven.
- Modification of an existing meter is supported on instances running custom pipeline model. For instances running Standard-match or IP-control pipeline model, you cannot modify an existing meter without the `OFPMBT_DSCP_REMARK` band, with a meter having `OFPMBT_DROP` band and vice-versa.

Meter statistics, scale, and limitations

Meter statistics

For meter statistics, you can use both aggregated and per band packet and byte counters.

Meter scale

- Instances running pipeline model such as standard-match or custom use meters from different pools.
- 2046 is the global limit for number of meters for instances running pipeline model such as standard-match and IP-control. All instances running in either of these two pipeline models use meters from this pool.



NOTE: The `openflow limit policy-engine-usage` command is used for determining the meter scale. By default, a total of 1,022 meters are available (default value of `policy-engine-usage` is 50%).

- The global limit of number of meters is 2,000 for instances running in custom pipeline model.

Meter limitations

- If the corresponding match does not have Ethernet type set to IPv4 or IPv6, the meter with a DSCP Remark meter band will be rejected.
- For standard-match and IP-control pipeline models:
 - Meters are not supported on software tables (200 to 203).
 - OpenFlow rules with **OFFPIT_METER** instruction having **OFFPMBT_DSCP_REMARK** band is not supported with **OFFPIT_GOTO_TABLE** and **OFFPAT_OUTPUT** to **OFFPP_CONTROLLER**.

Groups represent sets of actions for flooding as well as more complex forwarding semantics (for example, multipath, fast reroute, and link aggregation). As a general layer of indirection, groups also enable multiple flow entries to forward to a single identifier (for example, IP forwarding to a common next hop). This abstraction allows common output actions across flow entries to be changed efficiently.

The group table contains group entries; each group entry contains a list of action buckets with specific semantics dependent on group type. The actions in one or more action buckets are applied to packets sent to the group. There are four types of groups:

1. **All.** All the action buckets in the group are executed when a packet hits the group table.
2. **Select.** Execute any one action bucket in the group. The switch implementation uses round-robin to select the action bucket to be executed. OpenFlow specification defines a weight mechanism to do load sharing. However, this weight mechanism is not supported in the switch implementation. The weight **MUST** be given as 1. For all the other groups, weight **MUST** be specified as 0.
3. **Indirect.** Execute the one defined bucket in this group. This group supports only a single bucket.
4. **Fast failover.** Execute the first live bucket. The buckets are evaluated for liveness in the order defined by the group.

Group type ALL

The OpenFlow group type ALL is supported in both hardware and software. The capabilities are:

- Executes all the action buckets in the group.
- Supports groups in hardware, if each group bucket has only a single output port action.



NOTE: If a bucket has more than one instance of an output action or any other action, the bucket is installed on software instead of hardware.

- Virtual ports other than **OFPP_FLOOD** are not supported in hardware group buckets.
- Tunnel ports are not supported in group buckets of both hardware and software.

Group type SELECT

The OpenFlow group type SELECT is supported only in software. The capabilities are:

- Executes any one action bucket in the group.
- Uses round-robin to execute any one action bucket in the group.
- Defines weight mechanism for load sharing as per the OpenFlow specification. However, the weight mechanism is not available in the switch implementation. By default, set the weight as one for group type SELECT. For all other groups, the weight is zero.

Group Type INDIRECT

The OpenFlow group type INDIRECT is supported only in software. The capabilities are:

- Executes the one defined action bucket in the group.
- Supports only a single action bucket.

Group Type FAST FAILOVER

The OpenFlow group type FAST FAILOVER is supported only in software. The capabilities are:

- Executes the first LIVE action bucket in the group. The buckets are evaluated for liveness in the order defined by the group.
- OpenFlow virtual ports are not supported as watch ports.

Group actions

Supports the following actions in a group:

- `OFFPAT_OUTPUT`
- `OFFPAT_COPY_TTL_OUT` (software only)
- `OFFPAT_COPY_TTL_IN` (software only)
- `OFFPAT_SET_NW_TTL` (software only)
- `OFFPAT_DEC_NW_TTL` (software only)
- `OFFPAT_SET_FIELD` (software only)

Group statistics

The following statistics are supported in a group:

- Groups in software support both byte and packet counters.
- Groups in hardware support both byte and packet counters only on v3 modules.
- Groups in software also support bucket counters.

Group scale

The group scale limit is as follows:

- Supports a maximum of 1,024 groups across all OpenFlow instances.
- Supports a maximum of eight buckets per group.
- Supports a flow, which can point up to eight different groups (software groups and only one hardware group).

Group limitations

The following are the limitations in a group:

- For all group types other than FAST FAILOVER, the `watch_port` must be `OFPP_ANY`.
- Does not support `watch_group` in the group buckets other than `OFPG_ANY`
- A software group cannot point to a flow in the hardware.

- Does not support moving a hardware group after modification to a software group.
- If a flow on the hardware points to a group, the table cannot have any other associated actions.

OpenFlow supports per-flow rate-limiters for OpenFlow 1.0 as HPE vendor extensions.

A rate-limiter controls the rate of packets passing through a switch. Per-flow rate-limiters associate an arbitrary number of flows with a rate-limiter. Using OpenFlow with per flow rate-limiters, any number of flows can be flexibly mapped to a rate-limiter, regardless of their source and destination ports. Rate-limiters can be used via the HPE VAN SDN controller, which includes support for HPE QoS extensions. A limiter id (an arbitrary 32-bit number) addresses rate-limiters. Configuration of rate-limiters is done through a simple message from the controller, which can add, modify or remove a rate-limiter. Flows are directed to rate-limiters through an action. Multiple flows can be associated with the same rate-limiter. Statistics can be read from the OpenFlow controller for each rate-limiter.

**NOTE:**

Per-flow rate-limiters are used only if the hardware rate-limiter for the instance is disabled.

QoS extensions

HPE QoS extension to the OpenFlow protocol supports rate-limiters. A rate-limiter controls the rate of packets passed through it. Per-flow rate-limiters associate an arbitrary number of flows with a rate-limiter. The HPE QoS vendor extensions support per-flow rate-limiters only with the drop rate flag and not the remark rate or other flags.

Create a limiter

A per-flow rate-limiter is added/created via the HPE VAN SDN controller.

On receiving an OpenFlow message from the OpenFlow controller, the vendor ID is checked for a match to the HPE vendor ID. If the message type received indicated the addition of a new rate-limiter, a new rate-limiter is created with the parameters received in the message.

Get limiter details

The details on the limiters configured can be retrieved via the HPE VAN SDN controller. These details can also be checked on the switch CLI using the command `show openflow instance <instance-name> limiters`.

Support an OpenFlow flow with a limiter

A flow can be associated with a per-flow rate-limiter by giving the limiter ID as an OpenFlow action. For example, assume that a per-flow rate-limiter with ID 100 was created. From the HPE VAN SDN controller that supports HPE QoS extensions, a flow can be associated with this rate-limiter by using this limiter ID of 100 as one of the specified OpenFlow actions of that flow.

Overview

OpenFlow multi-VLAN instance associates multiple VLANs to an OpenFlow instance.

The capabilities of the OpenFlow multi-VLAN instance include:

- Ability to include more than one VLAN on the switch as member VLAN for an OpenFlow instance in Virtualized mode.
- Supported on both OpenFlow 1.0 and 1.3 versions.
- For a Multi-VLAN instance, the first VLAN in the Multi-VLAN instance is part of the DPID. Since VLANs cannot be part of multiple instances, this restriction ensures a unique DPID (the upper 16 bits of the DPID) for each instance.
- Packet-IN for a Multi-VLAN instance is the same as aggregate mode (that is, the Packet-IN is sent with the VLAN tag.) If the instance is a single VLAN instance, the behavior is the same as the virtualized instance (that is, the Packet-IN is sent untagged.)
- Ability to add OpenFlow rules with wild-carded VLAN in such an instance that matches traffic inbound on only those VLANs that are part of the instance.
- Ability to add OpenFlow rules with a specific VLAN that is part of the VLANs in the instance.
- Supports v2 and v3 modules.

Custom pipeline-model

The OpenFlow instance supports a pipeline-model type 'custom' on the v3 blades that allows the SDN controllers to create, modify, and destroy OpenFlow pipelines of its choice in hardware. The instance can be configured with its pipeline-model set to custom only if it is running in 'v3 mode'.

The capabilities provided by this pipeline-model include:

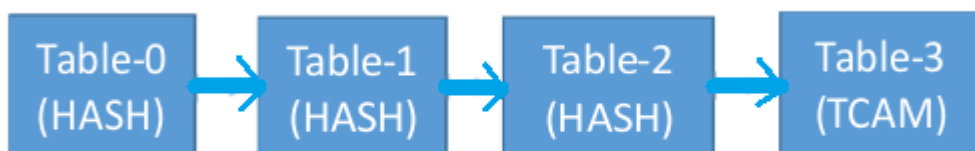
- SDN controller can create an OpenFlow pipeline with multiple flow tables, each with its own unique match and action capability.
- All the flow-tables in this pipeline-model will reside in hardware.
- Flow tables in this pipeline-model can be of two types:a. Tables with no wildcard and mask capabilities called HASH.b. Tables with wildcard and mask capabilities called TCAM.
- If the size of the tables requested by controller cannot be accommodated in hardware, the request is rejected.
- A maximum of 12 tables can be configured per instance by the controller.
- The minimum number of flows in a custom HASH table is 16.
- The minimum number of flows in a custom TCAM table is 2.

See **Flow table capabilities** on page 122 for the match and action capabilities supported in this pipeline-model.

Default Pipeline in custom mode

An instance can be configured in this mode via CLI by using the command `pipeline-model custom`.

Figure 13: Table pipeline



The switch consisting of three Hash tables and one TCAM table, as shown in Figure 5, advertises the default pipeline in custom mode. The default pipeline consists of four tables, three of type HASH and one of type TCAM with the following capabilities:

Table ID	0	1	2	3
Table Name	Custom L2 Src	Custom L2 Dst	Custom L3 Table	Custom TCAM Table
Metadata Match	0x0000000000000000 00	0x0000000000000000 00	0x0000000000000000 00	0x0000000000000000 00
Metadata Write	0x0000000000000000 00	0x0000000000000000 00	0x0000000000000000 00	0x0000000000000000 00
Max Entries	8k	8k	8k	2k
Match	ETH_SRC VLAN_VID	ETH_DST VLAN_VID	ETH_TYPE VLAN_VID IP_PROTO IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST IPV6_SRC IPV6_DST	IN_PORT ETH_DST ETH_SRC ETH_TYPE VLAN_VID VLAN_PCP IP_DSCP IP_PROTO IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST IPV6_SRC IPV6_DST

Table Continued

Table ID	0	1	2	3
Wildcards				IN_PORT ETH_DST ETH_SRC ETH_TYPE VLAN_VID VLAN_PCP IP_DSCP IP_PROTO IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST IPV6_SRC IPV6_DST
Instructions	GOTO APPLY WRITE CLEAR METER	GOTO APPLY WRITE CLEAR METER	GOTO APPLY WRITE CLEAR METER	APPLY WRITE CLEAR METER
Instructions Miss	GOTO APPLY WRITE CLEAR METER	GOTO APPLY WRITE CLEAR METER	GOTO APPLY WRITE CLEAR METER	APPLY WRITE CLEAR METER
Next Tables	1, 2, 3	2, 3	3	
Next Tables Miss	1, 2, 3	2, 3	3	

Table Continued

Table ID	0	1	2	3
Write Action	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL
Write Action Set-Field	ETH_DST ETH_SRC VLAN_VID VLAN_PCP	ETH_DST ETH_SRC VLAN_VID VLAN_PCP	ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST	ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST
Write Action Miss	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL

Table Continued

Table ID	0	1	2	3
Write Action Miss Set-Field	ETH_DST ETH_SRC VLAN_VID VLAN_PCP	ETH_DST ETH_SRC VLAN_VID VLAN_PCP	ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST	ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST
Apply Action	OUTPUT GROUP	OUTPUT GROUP	OUTPUT GROUP	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL
Apply Action Set-Field				ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST

Table Continued

Table ID	0	1	2	3
Apply Action Miss	OUTPUT GROUP	OUTPUT GROUP	OUTPUT GROUP	OUTPUT GROUP SET-FIELD PUSH_VLAN POP_VLAN IP_TTL
Apply Action Miss Set-Field				ETH_DST ETH_SRC VLAN_VID VLAN_PCP IP_DSCP IPV4_SRC IPV4_DST TCP_SRC TCP_DST UDP_SRC UDP_DST



NOTE: Only one instance will be operationally up in custom pipeline-model by default. The other instances running the custom pipeline-model will be operationally down with the reason as `Resources not available`. This is because all hardware resources are allocated to the first OpenFlow instance enabled with this pipeline-model. To bring up other instances running this pipeline-model, the SDN controller will have to free up resources in the existing pipeline on the first instance.

Pipeline modification process

Pipeline modification follows this basic process:

- An instance must be configured in the custom mode so that the controller can use the table modifications feature.
- The controller using the “Table Features” request message, triggers a table modification request.
- The first “OFPM_TABLE_FEATURES” is sent with an empty body and the switch replies with the default table pipeline.
- The controller understands the current pipeline.
- If the controller wants to modify the default table pipeline, it again sends “OFPM_TABLE_FEATURES” with the required table pipeline configuration.

- The table pipeline has details of the table numbers, size of each table, instructions/match/action capabilities of each table.
- If the table pipeline with all its properties and scale can be achieved in the hardware, the table modification request is accepted.
- The table modification can happen dynamically on custom pipeline instances (that is, instance need not have to be disabled to do the table modification).

Figure 14: *OpenFlow pipeline*

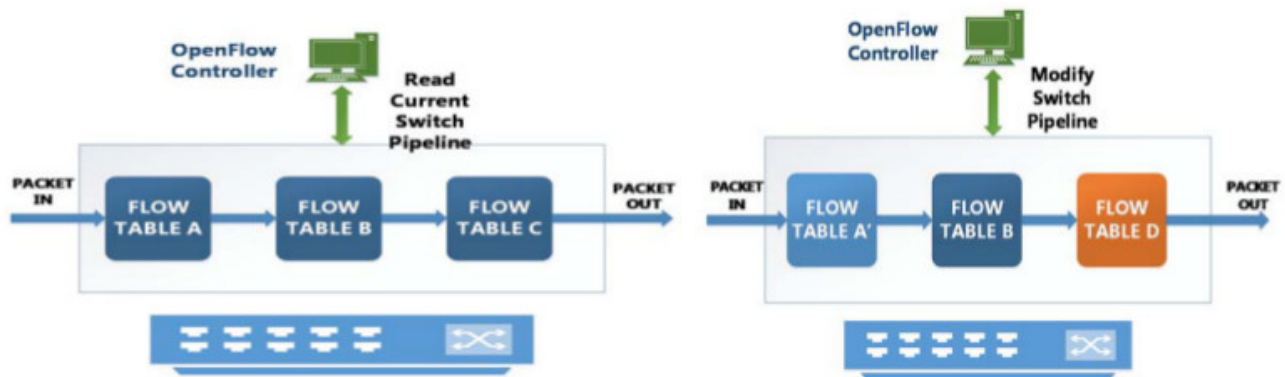
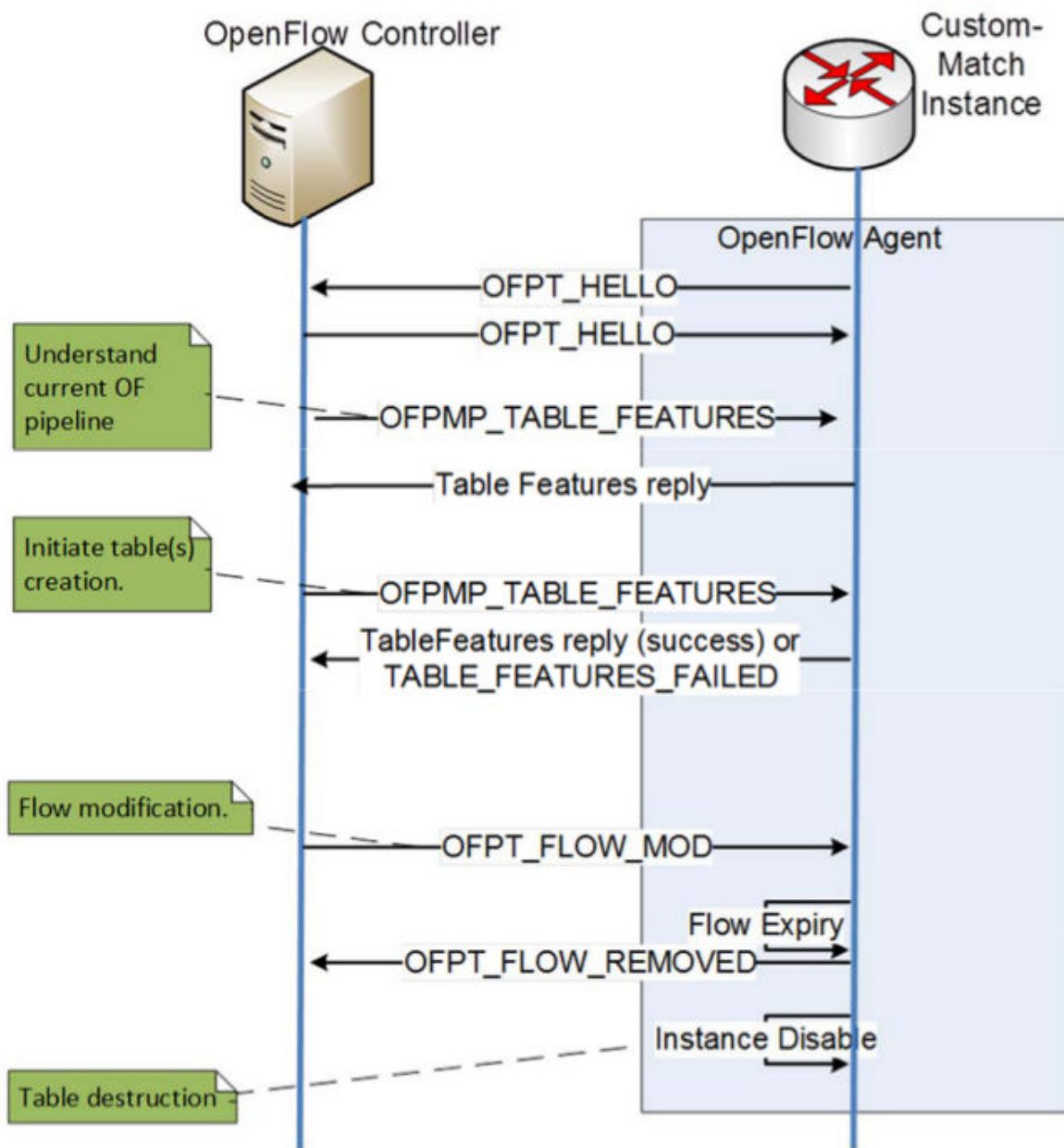


Figure 15: *OpenFlow custom match instance*



Example: Custom Pipeline Instance

```
switch(openflow)# show running-config
```

Running configuration:

```
; J9850A Configuration Editor; Created on release #KB.16.02.0000x
; Ver #0d:33.ff.7c.5f.fc.7b.ff.ff.fc.ff.ff.3f.ef:49
hostname "switch-name"
module A type j9991a
module F type j9987a
snmp-server community "public" unrestricted
openflow
  controller-id 1 ip 10.20.30.42 controller-interface vlan 2
  instance aggregate
```

```

        listen-port
        controller-id 1
        version 1.3 only
        pipeline-model custom
        enable
        exit
    enable
    exit
oobm
    ip address dhcp-bootp
    exit
vlan 1
    name "DEFAULT_VLAN"
    no untagged A2-A3,A5
    untagged A1,A4,A6-A24,F1-F24
    ip address dhcp-bootp
    exit
vlan 2
    name "VLAN2"
    untagged A2
    ip address 10.20.30.40 255.255.255.0
    exit
vlan 3
    name "VLAN3"
    untagged A3,A5
    no ip address
    exit
no allow-v2-modules

```

```
switch(openflow)# show openflow
```

```

OpenFlow           : Enabled
Egress Only Ports Mode : Disabled

```

Instance Information

Instance Name	Oper. Status	No. of H/W Flows	No. of S/W Flows	OpenFlow Version
aggregate	Up	8	0	1.3 only

```
switch(openflow)# show openflow instance aggregate
```

```

Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : aggregate
Data-path Description       : aggregate
Administrator Status       : Enabled
Member List                : VLAN 1, 3
Pipeline Model              : Custom Pipeline
Listen Port                : 6633
Operational Status         : Up
Operational Status Reason  : NA
Datapath ID                : 000140a8f09e8600
Mode                       : Active
Flow Location               : Hardware Only
No. of Hardware Flows      : 8
No. of Software Flows      : 0
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps
Conn. Interrupt Mode       : Fail-Secure
Maximum Backoff Interval   : 60 seconds
Probe Interval             : 10 seconds
Hardware Table Miss Count   : NA

```

```

No. of Software Flow Tables : NA
Egress Only Ports           : None
Table Model                 : Custom Pipeline
Source MAC Group Table      : Disabled
Destination MAC Group Table  : Disabled

```

Controller Id	Connection Status	Connection State	Secure	Role
1	Connected	Active	No	Equal

```
switch(openflow)# show openflow instance aggregate flow-table
```

OpenFlow Instance Flow Table Information

Table ID	Table Name	Flow Count	Miss Count	Goto Table
0	Custom L2 Src	1	0	1, 2, 3
1	Custom L2 Dst	1	0	2, 3
2	Custom L3 Table	1	0	3
3	Custom TCAM Table	5	0	*

Table ID	Table Name	Available Free Flow Count
0	Custom L2 Src	Slot A : 7372 Slot F : 7372
1	Custom L2 Dst	Slot A : 6144 Slot F : 6144
2	Custom L3 Table	Slot A : 5529 Slot F : 5529
3	Custom TCAM Table	Slot A : 2010 Slot F : 2010

* Denotes that the pipeline could end here.

Table 4: *OpenFlow match field groups*

Group 1	Group 2	Group 3	Group 4	Group 5	Group6	Custom(NOTE: Requires a separate Group for every field in this list)
ETH_DST	IPV4_SRC	IPV4_DST	VLAN_ID	TCP_SRC_RANGE	VLAN_UNTAGGED (Used only if the table has wildcards or maskable matches; has match on VLAN_VID, and adding this does not exceed the maximum limit of match groups in a table)	ARP_OP
ETH_SRC	IPV6_SRC	IPV6_DST	VLAN_PCP	TCP_DST_RANGE	VLAN_TAGGED (Used only if the table has wildcards or maskable matches; has match on VLAN_VID, and adding this to a table does not exceed the maximum limit of match groups in a table)	ARP_SPA
ETH_TYPE			IP_DSCP	UDP_SRC_RANGE		ARP_TPA
			IP_PROTO	UDP_DST_RANGE		ARP_SHA
			TCP_SRC			ARP_THA
			TCP_DST			ICMPV4_TYPE

Table Continued

Group 1	Group 2	Group 3	Group 4	Group 5	Group6	Custom(NOTE: Requires a separate Group for every field in this list)
			UDP_SRC			ICMPV4_CODE
			UDP_DST			IPV6_FLABEL
			IN_PORT (Used if none of the previous tables have an APPLY (set-field/ push_vlan/ pop_vlan)			ICMPV6_TYPE
						ICMPV6_CODE
						IPV6_ND_TARGE T
						TCP_FLAGS
						CUSTOM_ONE
						CUSTOM_TWO
						CUSTOM_THREE
						CUSTOM_FOUR
						IN_PORT (Used if any of the previous tables have an APPLY (set-field/ push_vlan/ pop_vlan)

Pipeline creation guidelines

Best practices to create a pipeline are:

- Number of tables in your pipeline must not be more than 12.
- Check if the table type has at least one wildcard-able field or a mask-able field.

- If `yes`, maps to a TCAM table in hardware.
- If `no`, maps to a HASH table in hardware.
- Determine the table type for each table in the pipeline. Once you identify the table type as HASH or TCAM, do the following:
 1. Verify how many match groups can be created for a table.
 2. Determine how many groups the matches the selected table type.
 3. Determine the number of tiles that will be consumed from the global pool to satisfy the scale of the requested table type.
- The minimum table size for a HASH table type is 16 and two for a TCAM table.
- The prerequisite fields defined by the OpenFlow specification have to be part of the table match field list. For example, you cannot create a table with match on `IPV4_SRC` address without adding `ETH_TYPE` as a match field.
- The first table in the pipeline must have the Table-ID as zero. The last valid Table-ID is 254.
- Ensure no references (flows with Go To instruction) to a table are being modified or deleted from a table (that is not updated), when you selectively modify the current pipeline doing any of the following:
 - Adding an additional table to the pipeline
 - Removing an existing table from the pipeline
 - Modifying the property of an existing table
 - All of the above

Performance in custom pipeline model

The performance of a custom pipeline model is based on factors such as table type, number of lookups, packet size, OpenFlow instruction type, and stream type.

Table Type

The following parameters impact the performance:

- A **miss** is more expensive on a HASH versus a TCAM table type.
- A **hit** is more expensive on a TCAM versus a HASH table type.

Number of lookups

The performance is reduced when a packet has to do more table lookups, before it egresses the pipeline.

Packet size

The packet size has a major impact on the performance. There is a limit on the number of concurrent packets the custom pipeline can service at a time. Thus, if the packet size is larger, the throughput in terms of `bps` is better.

Packets are dropped at the ingress, if the custom pipeline is busy servicing too many packets. You can check the RX discards on the ingress port to obtain such data.

OpenFlow instruction type

Flows with **APPLY** instruction yield low performance because of immediate action on the packet against the **WRITE** instruction, which accumulates all actions at the end of the pipeline.

Stream type

The traffic generated from the same source lowers the performance when compared to a more distributed stream.

Scale on custom pipeline Instance

- The hardware can match up to four different Groups of match in the custom-pipeline mode.
- The number of groups depends on the match fields in the HASH or TCAM table.

HASH table

- A single hash table can have match fields from any of the above three groups only.
- Based on the number of groups matched, the scale varies.
 - If the hash table is created with match fields from one group, it can hold up to 64k entries.
 - If the hash table is created with match fields from two or three groups, it can hold up to 32k entries.



NOTE: The maximum numbers of flows for 2930F/2930M switches are limited to 16k entries.

TCAM table

- A single tcam table can have match fields from up to four groups. The new group table has more than four groups.
- The scale varies depending on the number of groups.
 - If the tcam table is created with match fields from one group, it can hold up to 8k entries.
 - If the tcam table is created with match fields from two groups, it can hold up to 4k entries
 - If the tcam table is created with match fields from three or four groups, it can hold up to 2k entries



NOTE: An instance without “pipeline-model” configured defaults to standard-match but changes to IP-control mode when ip-control-mode is globally enabled.

Custom matches

The OpenFlow 1.3 specification defines a fixed set of packet header fields as OXM fields that can be used by the SDN controller to identify which fields in the packet it wants to match in any given flow on a flow table. The header fields defined are a list of well-known and popular protocol fields. The list is not exhaustive. There are still many packet fields that are not covered in the standard set but might be required to solve potential SDN use cases.

The SDN controller now can define one or more abstract match fields (termed **custom match**) in a flow-table of an instance running in custom pipeline-model when defining the new pipeline

Defining a custom match field

To use the customizable matches feature, the SDN controller first needs to create an OpenFlow table wherein it needs to define what these customizable fields will match on. The table can match on standard OpenFlow OXM match fields along with the customizable match fields. To create an OpenFlow pipeline with such tables, the controller sends an OFPMP_TABLE_FEATURES multipart request to the switch with a body that defines what the pipeline should look like.

OFPTFPT_MATCH is the table feature property type that needs to be set to tell the switch which match fields the SDN controller wants to match on for a given table. OFPTFPT_MATCH is an array of OXM headers of packet fields that will be matched on the table.

To define a customizable match field, the OXM header in the OFPTFPT_MATCH should look like the following:

0	1	2	3	4	5	6	7
OXM_CLASS		OXM_FIELD	OXM_LEN	EXPERIMENTER_ID			
START_TYPE		OFFSET		NUM_BYTES			

OXM_CLASS

This needs to be OFPXMC_EXPERIMENTER (0xffff).

OXM_FIELD

This should be one of the new OXM fields that have been defined for this purpose. It can take one of the following values:

- CUSTOM_MATCH_ONE = 5
- CUSTOM_MATCH_TWO = 6
- CUSTOM_MATCH_THREE = 7
- CUSTOM_MATCH_FOUR = 8

OXM_LEN

The length of the experimenter header excluding the OXM header.

EXPERIMENTER ID

The HPE VENDOR ID.

START TYPE

This field specifies where in the packet to look for the data. It can have one of the following values:

- L2_START = 1 /* Look from L2 header */
- L3_START = 2 /* Look from L3 header */
- L4_START = 3 /* Look from L4 header */

OFFSET

This field defines the byte offset from the 'START TYPE' where the match field will start.

NUM_BYTES

This field defines the number of bytes to match from the 'OFFSET'.

The following table shows an example to define a custom match field to match on TCP sequence number. All the data has to be in Network Byte Order.

0	1	2	3	4	5	6	7
0xffff (OFPXMC_EXPERIMENTER)		0x05(OXM_FIELD)	0x0a(OXM_LEN)	0x00002481 (HP VENDOR ID)			
0x0003 (Start at L4)		0x0004 (at 4 bytes offset)		0x0004 (match 4 bytes)			

Programming a flow with match on custom match field

Once the table with one or more custom match field is created on the switch, the controller can program flows to match on these fields via OFPT_FLOW_MOD message.

The data for the custom field will be embedded as an OXM field within the flow mod request message:

0	1	2	3	4	5	6	7
OXM_CLASS		OXM_FIELD	OXM_LEN	EXPERIMENTER_ID			
LENGTH OF MATCH DATA				MATCH DATA			
MATCH DATA							
MATCH DATA							

OXM CLASS

This needs to be OFPXMC_EXPERIMENTER (0xffff).

OXM FIELD

This should be the OXM fields for which the data is being defined:

- CUSTOM_MATCH_ONE = 5
- CUSTOM_MATCH_TWO = 6
- CUSTOM_MATCH_THREE = 7
- CUSTOM_MATCH_FOUR = 8

LENGTH

The length of the experimenter header excluding the OXM header.

EXPERIMENTER ID

The HPE VENDOR ID.

LENGTH OF DATA

This field defines the number of bytes of data to consider as the value for the custom match field in the following 16 bytes of data.

MATCH DATA

This field holds the data to look in the packet in the custom match field. Based on the value defined in the 'LENGTH OF DATA' field, only that many bytes are considered relevant from the MSB. The rest of the field needs to be padded with zeros

The following table shows an example of defining a custom match field to match on TCP sequence number with value of 0x12345678. The data has to be in Network Byte Order.

0	1	2	3	4	5	6	7
0xffff (OXM_CLASS)		0x05 (OXM_FIELD)	0x18 (OXM_LEN)	0x00002481(HPE VENDOR ID)			
0x00000004 (Length of match data)				0x12345678 (Match data)			
0x0000000000000000							
0x00000000							

Facts

- We support up to four such customizable fields per OpenFlow table.
- The custom fields defined are local to that table. That is, the OXM field CUSTOM_MATCH_ONE...FOUR can map to a different packet field on each table. They are not global to the whole of OpenFlow pipeline.
- Each custom match field can match up to 128 bits of data. The fields can also be masked and wild carded.
- The maximum allowed offset from any start type is 63 bytes.
- A flow table can have a combination of standard OXM fields with custom match fields.
- Each custom match field consumes a match group in itself.

When the controller wishes to send out a packet through the OpenFlow switch, it uses the `OFPT_PACKET_OUT` message.

All the actions supported on the software tables of standard-match and IP Control pipeline models are supported in the `packet_out` message.

IN_PORT

- If the `in_port` in the `packet_out` message is set to `OFPP_CONTROLLER` or `OFPP_NONE`, the switch treats it as a CPU generated packet.
- If the `in_port` in the `packet_out` message is a logical port that is not a member of the OpenFlow instance for which the `packet_out` message is sent, the switch rejects the `packet_out` message.

OUT_PORT

- If the `out-port` in the `packet_out` is `OFPP_NORMAL`, the switch performs one of the following:
 - Attempts to route the packets if the Ethernet destination address is set to switch MAC address in the payload.
 - Attempts to bridge the packet for non-MAC address packets.
- If the `out-port` in the `packet_out` is `OFPP_LOCAL`, the switch sends the packet to the local IP stack in the following scenarios:
 - Packets with Ethernet destination address set to switch MAC address or broadcast MAC address in the payload.
 - Packet IP address is set to IP VLAN of the switch in a payload.
- Does not support the virtual port, `OFPP_TABLE`.

Determining the VLAN

The switch uses the following steps to determine the VLAN to be associated with the packet in the payload:

Procedure

1. Check the VLAN ID in the payload, if tagged.
2. Use the primary VLAN of the `in-port` for untagged packets.

If the `in-port` does not have a primary VLAN associated with it, the first instance of the VLAN in the list is used. The first instance of the VLAN in the list if the packet is untagged and `in-port` is a tunnel port.



NOTE:

- If the VLAN determined from the above step is not a member VLAN of the instance, the packet is dropped.
 - If the packet in the `packet_out` message is an LLDP frame, the packet is always sent untagged to the switch.
-

Matching on TCP flags and TCP/UDP port ranges

TCP Flags and L4 port ranges are supported as match parameters in Flow Modification messages from the controller. Experimenter Match Fields are used to accomplish this operation.

Table 5: *Matching on TCP Flags and TCP/UDP port ranges*

Supported	Not supported
Match on TCP Flags and UDP/TCP source/destination port ranges in OpenFlow rules.	TCP Flags and UDP/TCP source/destination port ranges cannot be set/modified using an OpenFlow action.
Match support available for v1.3 instances only.	Instances negotiated to 1.0 do not support the feature (as 1.0 does not support Experimenter Match fields).
Match capability is available for rules in Policy Tables (TCAM) and Software Tables.	IP-Control-Table (Table-50) and MAC Table (Table 40/41) does not support matching on TCP flags or port ranges.
Matching is supported on the previously mentioned tables for Instances configured in IP-Control table mode and Standard Match Mode.	Masking is not supported for port ranges.
Match masks are supported for TCP Flags.	N/A
Experimenter OXM class is used to communicate the match capability to the controller (in TABLE_FEATURES_REPLY).	N/A
Controllers wanting to program a rule with these match parameters must use the Experimenter Flow Match Fields to specify the TCP Flags/Port range match parameters along with the experimenter ID.	N/A

Experimenter match fields

ONF assigned vendor ID is expected in the experimenter Field for all the OpenFlow messages containing the Experimenter Match fields.

0x00002481 is the HPE Vendor ID.

Controllers in the “oxm_field” of the Experimenter Match Fields in Flow Modification messages must use the following values:

OXM field	Value
UDP Source Port Range	0
UDP Destination Port Range	1
TCP Source Port Range	2
TCP Destination Port Range	3
TCP Flags	4
Custom Match One	5
Custom Match Two	6
Custom Match Three	7
Custom Match Four	8

Values must be encoded post the Experimenter ID field in the OXM. For TCP Flags, the value and mask are each 16 bits in length. Similarly for Port Ranges, 16-bits are used to encode the Range Begin value followed by 16 bits for Range End value.

Flow Mod Validations

Table 6: *Flow Mod Validations*

Scenario	Error Type	Error Code
The “experimenter” field in the experimenter header of flow mod message contains a value other than HP_VENDOR_ID	OFFPET_BAD_MATCH	OFFBMC_UNKNOWN
The payload for the TCP flags match field contains an invalid value. (Example: in TCAM, if any bit in position 6:7 is set)	OFFPET_BAD_MATCH	OFFBMC_BAD_VALUE
If either OFFPXMT_OFB_TCP_SRC/ OFFPXMT_OFB_UDP_SRC or OFFPXMT_OFB_TCP_DST/ OFFPXMT_OFB_UDP_DST is already present in the flow mod and L4 source port range or, L4 destination port range matching is also specified respectively.	OFFPET_BAD_MATCH	OFFBMC_DUP_FIELD

Table Continued

Scenario	Error Type	Error Code
If the port range match is specified without the OFPXMT_OFB_IP_PROTO match pre-requisite.	OFPET_BAD_MATCH	OFPBMC_BAD_PREREQ
If the TCP flag matching is specified without the IP_PROTO match being set to 6 (TCP).	OFPET_BAD_MATCH	OFPBMC_BAD_PREREQ
If all the TCAM range registers for a slot are used up, an attempt to install a new rule on the slot with L4 port range match returns an OFPET_EXPERIMENTER error message to the controller. The Error code is as agreed between the Switch and the Controller team (0). The "experimenter" field has the HP_VENDOR_ID.	OFPET_EXPERIMENTER	Exp_type = OFPERR_OFPET_EXPERIMENTER_RANGE_FULL (0)

TCP Flags/L4 port range matching in Custom Match Mode



NOTE:

Matching of TCP Flags and L4 port ranges in Custom Match Mode are available from builds KB.16.01, KB.16.02 and WC.16.02 only.

Matching on TCP Flags and TCP/UDP port ranges in Custom Match Mode is supported using the match parameters `Experimenter Match Fields`.

Addition or modification of tag for single-tagged packets

This addition or modification is accomplished with the action "OFPAT_SET_FIELD" and setting the field to "OFPAT_SET_VLAN_VID (Set the 802.1q VLAN id)" inside the packet.

Is/Is-not table

IS	IS-NOT
TCP Flags match supported on TCAM and Hash tables in Custom match mode.	L4 Port Range matching is not supported in Hash Table in Custom match mode.
The lower 8 bits of TCP flags can be matched in TCAM and Hash tables in Custom match mode.	Masking of TCP flags is not supported in Hash tables.
Table Features request with TCP Flags and L4 Port Ranges as match fields is honored.	Default pipeline for Custom Match Mode does not support matching on TCP Flags or L4 Port ranges.

Restrictions

The default Custom Match Mode pipeline does **not** support matching on TCP Flags or L4 port ranges. The switch does, however, honor a "TABLE features" request message from the controller with match parameters in the `OFPTFPT_MATCH` property. The `oxm_class` for these match properties are `OFPXMC_EXPERIMENTER` and `oxm_field` values.

The following table shows the encoding format expected.

<code>oxm_class =</code> <code>OFPXMC_EXPERIMENTER</code>	<code>oxm_field =</code> <code>OFPXMT_OFEXP_UDP_SRC_PORT_RANGE/</code> <code>OFPXMT_OFEXP_UDP_DST_PORT_RANGE/</code> <code>OFPXMT_OFEXP_TCP_SRC_PORT_RANGE/</code> <code>OFPXMT_OFEXP_TCP_DST_PORT_RANGE/</code> <code>OFPXMT_OFEXP_TCP_FLAGS</code>
	Has Mask
	<code>oxm_length</code>
Experimenter ID (0x00002481)	

There are 60 Range resources available per v3 card. By default, OpenFlow is configured to use only 50% of the resources.

Each range match in a rule consumes one range resource, with the following exceptions:

- If a rule has same range values for L4 source and L4 dest range, two resources are consumed.
- If two rules have the same UDP/TCP source/Destination range respectively, only one range resource is consumed.

Range Resource Example

If Rule1 has TCP source=100-200 and Rule 2 has TCP Source=100-200, only 1 range resource is consumed. Similarly, if Rule1 has TCP source=100-200 and Rule2 has UDP source=100-200, 2 range resources are consumed. Also, a rule with TCP source=200-300 and TCP Destination=200-300 consumes 2 range resources.

Error Messages

The following error is applicable for TABLE FEATURES request message processing on the switch. This error code is applicable for custom match mode only. All other error codes mentioned in **LINK** are also applicable to Custom Match Mode.

Scenario	A Table Features Request Message from the controller containing individual L4 port matching along with L4 port range matching.
Error Type	OFFPET_TABLE_FEATURES_FAILED
Error Code	OFPTFFC_EPERM

OFPAT_PUSH_VLAN and OFPAT_POP_VLAN are supported for single tagged packets only.

Is	Is not
Support for OFPAT_PUSH_VLAN and OFPAT_POP_VLAN on all OpenFlow instance pipeline models.	OFPAT_PUSH_VLAN and OFPAT_POP_VLAN actions are not supported on v1 mode of operation.
Support for OFPAT_PUSH_VLAN and OFPAT_POP_VLAN on v2 and v3 modes of operation.	OFPAT_PUSH_VLAN action does not support double tagging.
Support for OFPAT_PUSH_VLAN only with ether-type as 0x8100 (802.1q)	OFPAT_PUSH_VLAN and OFPAT_SET_FIELD of OFP_VLAN_VID actions are not supported in the same flow.
OFPAT_PUSH_VLAN action on an already tagged packet does not add a fresh 802.1q tag. It modifies the existing 802.1q tag in the packet.	
OFPAT_POP_VLAN action strips only the 802.1q tag in the packet.	

Show OpenFlow information

You can display OpenFlow information for all instances, ports, and flows. The returned information includes the OpenFlow version supported.

Syntax

```
show openflow {auxiliary-connections | controllers | flow-table | instance <instance-name> | multiport-filter-limit | resource }
```

Show OpenFlow information.

auxiliary-connections

Show a list of OpenFlow auxiliary connections.

controllers

Show OpenFlow controller information.

flow-table

Display information for an OpenFlow flow table.

instance

Show information for an OpenFlow instance.

multiport-filter-limit

Show the multiport filter allocated to OpenFlow.

resources

Show OpenFlow resource utilization.

Show global OpenFlow information

Syntax

```
show openflow
```

Show OpenFlow information

Show openflow

```
switch# show openflow
OpenFlow           : Enabled
Egress Only Ports Mode : Disabled
```

Instance Information

Instance Name	Oper. Status	No. of H/W Flows	No. of S/W Flows	OpenFlow Version
titan	Up	6	4	1.3
marez	Up	6	4	1.3 only

Show auxiliary connection information

Only one auxiliary connection is supported per main controller connection.

Syntax

```
show openflow auxiliary-connections
```

Displays auxiliary connection information.

Example: Show OpenFlow auxiliary-connections

```
show openflow auxiliary-connections
```

```
Auxiliary
Conn. Index Type Port
-----
1          TCP  7777
2          UDP  8888
```

Show OpenFlow controllers

Displays OpenFlow controllers configured for use by OpenFlow.

Syntax

```
show openflow controllers
```

Example: show openflow controllers

```
switch(config)# show openflow controllers
```

```
Controller Information
```

```
Controller Id IP Address      Port   Interface
-----
1           20.0.0.2        6633   VLAN 6
```

Show OpenFlow flow table information

Syntax

```
show openflow flow-table
```

Displays global flow table information.

Show OpenFlow flow tables

```
switch# show openflow flow-table
```

```
Flow Table Information
```

```
Table Name           Max.   Refresh   Flow
                   Usage  Rate  (seconds) Count
-----
IP Control Table     50%    12         0
Policy Engine Table  50%    20         0
```

Slot ID	IP Control Table	Policy Engine Table
	Current Usage (%)	Current Usage (%)
1	0.000000	0.07
6	0.000000	0.07



NOTE: Current usage is percentage of OpenFlow maximum usage.

Show OpenFlow instance

Syntax

```
show openflow instance <instance-name>
```

Show OpenFlow instance information.

Show OpenFlow instance <instance-name>

```
switch(openflow)# show openflow instance test
Configured OF Version           : 1.3 only
Negotiated OF Version          : 1.3
Instance Name                   : test
Data-path Description           : test
Administrator Status            : Enabled
Member List                     : VLAN 3
Pipeline Model                  : Standard Match
Listen Port                     : 6633
Operational Status              : Up
Operational Status Reason       : NA
Datapath ID                     : 000340a8f09e8600
Mode                            : Active
Flow Location                   : Hardware and Software
No. of Hardware Flows           : 6
No. of Software Flows           : 4
Hardware Rate Limit             : 0 kbps
Software Rate Limit             : 100 pps
Conn. Interrupt Mode            : Fail-Secure
Maximum Backoff Interval        : 60 seconds
Probe Interval                  : 10 seconds
Hardware Table Miss Count       : NA
No. of Software Flow Tables     : 1
Egress Only Ports               : A1,A4,A6-A24,F1-F24
Table Model                     : Policy Engine and Software
Source MAC Group Table          : Disabled
Destination MAC Group Table     : Disabled
Controller Id Connection Status Connection State Secure Role
-----
1 Connected Active No Equal
```

show openflow instance test message-statistics

Syntax

```
switch# show openflow instance <INSTANCE-NAME> message-statistics
```

Description

Displays the descriptive reasons for the OpenFlow error codes which are returned to the controller for flow modifications and pipeline modifications.

Example

show openflow instance test message-statistics

```
switch# show openflow instance test message-statistics
OpenFlowMessage
Type                                     Received      Rejected
-----
OFPT_FLOW_MOD                          4             1
OFPT_PORT_MOD                          0             0
OFPT_GROUP_MOD                         0             0
OFPT_METER_MOD                         0             0
OFPMP_TABLE_FEATURES (Change Pipeline) 0             0
OFPMP_TABLE_FEATURES (View Pipeline)   0             0

Flow-Mod Error Counters
OpenFlow
ErrorCode                               Count          Reason
-----
OFPFMFC_EPERM                          1             Input port in the rule is not valid.

Pipeline-Mod Error Counters
OpenFlowError                           Code           Count Reason
-----
```

Viewing multiport-filter-limit

Syntax

show openflow multiport-filter-limit

Displays multiport filter information. (Only in OpenFlow version 1.3.)

Example: Viewing multiport filter information

```
switch# show openflow multiport-filter-limit

Total Multiport Filters: 1039

Features      Filters      Filters      Filters
Allocated    Used         Free
-----
OpenFlow      519          1           518
```

Show OpenFlow resources

Syntax

show openflow resources

Show OpenFlow resource usage in Policy Enforcement Engine.

Show openflow resources

```
switch(of-inst-test)# show openflow resources

Ingress Policy Enforcement Engine Rules
```


Resource usage in Policy Enforcement Engine

	Slots	Rules	Rules Used							
		Available	ACL	QoS	IDM	VT	Mirr	PBR	OF	Other
A		8158	0	0	0	0	0	0	14	4
F		8158	0	0	0	0	0	0	14	4

Ingress Policy Enforcement Engine Meters

	Slots	Meters	Meters Used							
		Available	ACL	QoS	IDM	VT	Mirr	PBR	OF	Other
A		2046		0	0	0			1	0
F		2046		0	0	0			1	0

Ingress Policy Enforcement Engine Port Ranges

Slots	Application Port Ranges	Application Port Ranges Used								
	Available	ACL	QoS	IDM	VT	Mirr	PBR	OF	Other	
A	60	0	0	0		0	0	0	0	
F	60	0	0	0		0	0	0	0	

Ingress Policy Enforcement Engine PBR Resources

Slots	PBR	PBR Next-hops Used								
	Next-hops Available	ACL	QoS	IDM	VT	Mirr	PBR	OF	Other	
A	1024						0		0	
F	1024						0		0	

6 of 32 Policy Engine management resources used.

Resource usage in Custom Pipeline Engine

Tables Available		Tables Used
A	32	0
F	32	0
Action-Set Available		Action-Set Used
A	32000	0
F	32000	0
Meters Available		Meters Used
A	2000	0
F	2000	0
Counters Available		Counters Used
A	31000	0
F	31000	0
Ranges Available		Ranges Used
A	64	0
F	64	0

Key:

ACL = Access Control Lists

QoS = Device & Application Port Priority, QoS Policies, ICMP rate limits
 IDM = Identity Driven Management
 VT = Virus Throttling blocks
 Mirr = Mirror Policies, Remote Intelligent Mirror endpoints
 PBR = Policy Based Routing Policies
 OF = OpenFlow
 Other = Management VLAN, DHCP Snooping, ARP Protection, Jumbo IP-MTU,
 RA Guard, Control Plane Protection, Service Tunnel, ND Snooping, UWW,
 mDNS, tunneled-node-server.

Resource usage includes resources actually in use, or reserved for future use by the listed feature. Internal dedicated-purpose resources, such as port bandwidth limits or VLAN QoS priority, are not included.



NOTE: Resource usage includes resources actually in use, or reserved for future use by the listed feature. Internal dedicated-purpose resources, such as port bandwidth limits or VLAN QoS priority, are not included.

Show OpenFlow instance group

Syntax

```
show openflow instance <instance-name> groups
```

Show openflow instance group

```
switch(openflow)# show openflow instance test groups
```

```
OpenFlow Instance Groups
```

```
  Group ID           : 2
  Group Type         : All
  Group Location      : Hardware
  Is Hardware Counter Available : Yes
  Reference Count     : 0
  Packet Count       : 0
  Byte Count          : 0
  Duration            : 57
  Action Buckets      : 1, 2
```

```
    Bucket 1
```

```
      Packet Count    : NA
      Byte Count       : NA
      Watch port       : Any
      Weight           : 0
      Action           : Output F12
```

```
    Bucket 2
```

```
      Packet Count    : NA
      Byte Count       : NA
      Watch port       : Any
      Weight           : 0
      Action           : Output F23
```

```
switch(openflow)# show openflow instance test groups
```

```
OpenFlow Instance Groups
```

```
  Group ID           : 3
  Group Type         : All
  Group Location      : Software
  Reference Count     : 0
  Packet Count       : 0
  Byte Count          : 0
```

```

Duration                : 27
Action Buckets          : 1
  Bucket 1
    Packet Count        : 0
    Byte Count          : 0
    Watch port          : Any
    Weight               : 0
    Action               : Output A2
                        : Output A3

```

Show OpenFlow instance flow table information

These commands display per instance flow table information, including both hardware and software flow tables.



NOTE: This option is available only for instances running OpenFlow version 1.3.

Viewing specific table capability

Syntax

```
show openflow instance <instance-name> flow-table<table-id> table-capability
```

Shows OpenFlow table capability information for a specific flow table ID.

Display OpenFlow flowtable capability

```
switch(of-inst-test)# show openflow instance test flow-table 100 table-capability
```

OpenFlow Flow Table Properties

Table Match Capabilities:

Incoming Port	Ethernet Destination
Ethernet Source	Ethernet Type
VLAN ID	VLAN PCP
IP DSCP	IP Protocol
IPv4 Source Address	IPv4 Destination Address
TCP Source Port	TCP Destination Port
UDP Source Port	UDP Destination Port
ICMP Type	ICMP Code
IPv6 Source Address	IPv6 Destination Address
UDP Source Port Range	UDP Destination Port Range
TCP Source Port Range	TCP Destination Port Range
TCP Flags	

Table Wildcard Capabilities:

Incoming Port	Ethernet Destination
Ethernet Source	Ethernet Type
VLAN ID	VLAN PCP
IP DSCP	IP Protocol
IPv4 Source Address	IPv4 Destination Address
TCP Source Port	TCP Destination Port
UDP Source Port	UDP Destination Port
ICMP Type	ICMP Code
IPv6 Source Address	IPv6 Destination Address
UDP Source Port Range	UDP Destination Port Range
TCP Source Port Range	TCP Destination Port Range
TCP Flags	

Table Instructions:

Metering

```

Apply Actions
  Set-Field
    Ethernet Destination
    VLAN ID
    IP DSCP
  Output
    Push VLAN
    Pop VLAN
GoTo 200
Table-Miss Instructions:
  Metering
  Apply Actions
    Output
    GoTo 200

```

```

Ethernet Source
VLAN PCP
Push VLAN
Group

```

```

Group

```

```
switch(of-inst-test)# show openflow instance test flow-table 200 table-capability
```

OpenFlow Flow Table Properties

Table Match Capabilities:

Incoming Port	Ethernet Destination
Ethernet Source	Ethernet Type
VLAN ID	VLAN PCP
IP DSCP	IP ECN
IP Protocol	IPv4 Source Address
IPv4 Destination Address	TCP Source Port
TCP Destination Port	UDP Source Port
UDP Destination Port	ICMP Type
ICMP Code	ARP Opcode
ARP Source IPv4 Address	ARP Target IPv4 Address
ARP Source Hardware Address	ARP Target Hardware Address
IPv6 Source Address	IPv6 Destination Address
IPv6 Flow Label	ICMPv6 Type
ICMPv6 Code	Target Address for ND
Source Link-Layer for ND	Target Link-Layer for ND
IPv6 Extension Header Pseudo-field	
UDP Source Port Range	UDP Destination Port Range
TCP Source Port Range	TCP Destination Port Range
TCP Flags	

Table Wildcard Capabilities:

Incoming Port	Ethernet Destination
Ethernet Source	Ethernet Type
VLAN ID	VLAN PCP
IP DSCP	IP ECN
IP Protocol	IPv4 Source Address
IPv4 Destination Address	TCP Source Port
TCP Destination Port	UDP Source Port
UDP Destination Port	ICMP Type
ICMP Code	ARP Opcode
ARP Source IPv4 Address	ARP Target IPv4 Address
ARP Source Hardware Address	ARP Target Hardware Address
IPv6 Source Address	IPv6 Destination Address
IPv6 Flow Label	ICMPv6 Type
ICMPv6 Code	Target Address for ND
Source Link-Layer for ND	Target Link-Layer for ND
IPv6 Extension Header Pseudo-field	
UDP Source Port Range	UDP Destination Port Range
TCP Source Port Range	TCP Destination Port Range
TCP Flags	

Table Instructions:

Apply Actions	
Set-Field	
Ethernet Destination	Ethernet Source
Ethernet Type	VLAN ID
VLAN PCP	IP DSCP

IP ECN	IP Protocol
IPv4 Source Address	IPv4 Destination Address
TCP Source Port	TCP Destination Port
UDP Source Port	UDP Destination Port
ICMP Type	ICMP Code
ARP Opcode	ARP Source IPv4 Address
ARP Target IPv4 Address	ARP Source Hardware Address
ARP Target Hardware Address	IPv6 Source Address
IPv6 Destination Address	IPv6 Flow Label
ICMPv6 Type	ICMPv6 Code
Target Address for ND	Source Link-Layer for ND
Target Link-Layer for ND	

Output

Show OpenFlow instance information

You can display OpenFlow instance information.

Syntax

```
show openflow instance <instance-name>
```

Show OpenFlow instance information.

capabilities

Show OpenFlow capabilities exchanged with the controller.

flow-table

Display information for a flow table.

flows

Show flow entries.

groups

Show group table information.

limiters

Show OpenFlow rate limits.

message-statistics

Show message statistics information for an instance.

meters

Show instance-specific meters.

port-statistics

Show port statistics.

Show OpenFlow instance capabilities information

Syntax

```
show openflow instance <instance-name> capabilities
```

Displays OpenFlow instance capabilities.

Example: Display OpenFlow instance capabilities information

```
switch# show openflow instance test capabilities

Policy Engine Match Capability : Extended Match

Switch Capabilities
-----
Flow Statistics
Table Statistics
Port Statistics
Group Statistics
Blank Ports
```

Viewing OpenFlow instance flow-table

Syntax

```
show openflow instance <instance-name> flow-table
```

Show flow table

```
switch(of-inst-test)# show openflow instance test flow-table
```

OpenFlow Instance Flow Table Information

Table ID	Table Name	Flow Count	Available Flow Count	Free Count	Miss Count	Goto Table
0	Start	1	NA		0	100
100	Policy Table	5	NA		0	200
200	SW Table 1	4	NA		0	*

* Denotes that the pipeline could end here.

Show OpenFlow instance flows

Shows the flow table entries for a particular OpenFlow instance.

Syntax

```
show openflow instance <instance-name> flows {flow-type}
```

flow-type shows the flow table entries for a particular OpenFlow instance.

The various flows that can be shown using the **flow-type** are:

destination-ip

Show flows matching the destination IP address.

destination-mac

Show flows matching the destination MAC address.

destination-port

Show flows matching the destination port.

ethernet-type

Show flows matching the EtherType.

ip-protocol

Show flows matching the IP protocol.

ip-tos-bits

Show flows matching the IP ToS bits.

source-ip

Show flows matching the source IP address.

source-mac

Show flows matching the source MAC address.

source-port

Show flows matching the source port.

vlan-id

Show flows matching the VLAN ID.

vlan-priority

Show flows matching the VLAN priority.

destination-ipv6

Show flows matching the destination IPv6 address.

flow-table

Show only flows that are present in the flow-table mentioned.

ingress-port

Show flows matching the ingress port.

source-ipv6

Show flows matching the source IPv6 address.

Example: Flow version 1.0

```
(<openflow>)# show openflow instance titan flows
Flow 1 Match
  Incoming Port      : F24                      Ethernet Type      : IP
  Source MAC         : 000000-000000             Destination MAC    : 000000-000000
  VLAN ID            : 0                         VLAN Priority       : 0
  Source Protocol Address : 255.255.255.255/32
  Target Protocol Address : 128.128.128.128/32
  IP Protocol         : 0x00                     IP ToS Bits        : 0
  Source Port         : 0                         Destination Port    : 0
Attributes
  Priority            : 32768                      Duration           : 10 secs
  Hard Timeout        : 0 secs                     Idle Timeout       : 60 secs
  Byte Count          : 0                          Packet Count       : 0
  Controller ID       : 1                          Cookie             : 0x0
  Flow Location        : Software                   Hardware Index     : 1
  Reason Code         : 100
  Reason Description   : Rule is in hardware
Actions
  Modify Destination IP : 183.23.45.64
  Modify Source IP      : 200.123.23.54
  Output                : A21
```

Example: Flow version 1.3

```

(<openflow>)# show openflow instance titan flows
Flow 1
Match
  Incoming Port      : 1/17
  Source MAC         : 000000-000000
  VLAN ID            : 0
  Source Protocol Address : 255.255.255.255/32
  Target Protocol Address : 128.128.128.128/32
  IP Protocol        : TCP
  IP ECN              : 0
  Source Port        : 0
  Ethernet Type       : IP
  Destination MAC     : 000000-000000
  VLAN Priority       : 0
  IP ToS Bits         : 0
  IP DSCP              : 18
  Destination Port    : 0
Attributes
  Priority            : 32768
  Hard Timeout       : 0 secs
  Byte Count         : 5040
  Flow Table ID      : 3
  Activity Count     : 0xffffffff
  Hardware Index     : 1
  Duration           : 10 secs
  Idle Timeout       : 0 secs
  Packet Count       : 28
  Controller ID      : 1
  Cookie             : 0x0
Instructions
  Clear Actions
    Write Actions
      Pop VLAN
      Push VLAN Decrement TTL
      Output: : 3/24, 4/5, 1/18
      Goto Table ID: 2
Flow 2
Match
  Incoming Port      : Trk1
  Source MAC         : 000000-000000
  VLAN ID            : 0
  Source Protocol Address : 255.255.255.255/32
  Target Protocol Address : 128.128.128.128/32
  IPv6 Flow Label     : 0
  IPv6 Ext. Header    : Fragment
  ND Source MAC       : 000000-000000
  ND Target IP        : 0:0:0:0:0:0:0:0
  IP Protocol         : 0x2C
  IP ECN              : 0
  Source Port        : 0
  Ethernet Type       : IPv6
  Destination MAC     : 000000-000000
  VLAN Priority       : 0
  Destination MAC     : 00000-000000
  IP DSCP              : 20
  Destination Port    : 0
Attributes
  Priority            : 12345
  Hard Timeout       : 300 secs
  Byte Count         : 0
  Flow Table ID      : 6
  Activity Count     : 0xffffffff
  Hardware Index     : 1
  Duration           : 10 secs
  Idle Timeout       : 160 secs
  Packet Count       : 0
  Controller ID      : 1
  Cookie             : 0x0
Instructions
  Apply Actions
    Modify Destination IP : 2000::5
    Modify Source IP      : 2000::6
    Modify Source MAC     : 121212-121212
    Modify Destination MAC : 131313-131313
    Modify VLAN ID        : 123
    Modify IP DSCP         : 18
    Modify IP ECN          : 1
    Decrement TTL
    Meter ID              : 112
    Group ID              : 2
  Write Actions
    Decrement TTL
    Goto Table ID        : 4
Flow 3
Match
  Incoming Port      : 0
  Source MAC         : 000000-000000
  Ethernet Type       : ARP
  Destination MAC     : 000000-000000

```



```

VLAN ID          : 0          VLAN Priority      : 0
ARP Opcode       : 1
ARP Source MAC   : 00A0C9-22B210   ARP Target MAC : 000000-000000
Source Protocol Address : 255.255.255.255/32
Target Protocol Address : 128.128.128.128/32
Attributes
Priority          : 32768          Duration       : 10 secs
Hard Timeout     : 0 secs         Idle Timeout    : 0 secs
Byte Count       : 12450          Packet Count    : 2323
Flow Table ID    : 3             Controller ID   : 3
Activity Count   : 0xffffffff     Cookie          : 0x0
Hardware Index   : 1
Flow 4
Match
Source MAC       : 000000-000000   Ethernet Type   : 0x8035
VLAN ID          : 0             Destination MAC : 000000-000000
ARP Opcode       : 0             VLAN Priority    : 0
ARP Source MAC   : 000000-000000
ARP Target MAC   : 000000-000000
Source Protocol Address : 0.0.0.0
Target Protocol Address : 0.0.0.0
Source IP        : 0.0.0.0
Destination IP   : 0.0.0.0       ARP Target IP    : 0.0.0.0
IPv6 Flow Label  : 0
IPv6 Ext. Header : None
ND Source MAC    : 000000-000000
ND Target IP     : 0:0:0:0:0:0:0:0 ND Destination MAC : 000000-000000
IP Protocol      : 0x00
IP ECN           : 0             IP DSCP          : 0
Source Port      : 0             Destination Port  : 0
Attributes
Priority          : 32768          Duration       : 15 secs
Hard Timeout     : 0 secs         Idle Timeout    : 50 secs
Byte Count       : 0             Packet Count    : 0
Flow Table ID    : 5             Controller ID   : 5
Activity Count   : 0xffffffff     Cookie          : 0x0
Hardware Index   : 1
Instructions
Write Actions
  Output : 2/1
  Output : Controller

```

Show group information for a specific instance

Syntax

```
show openflow instance <instance-name> groups
```

Displays group information for a specific instance.

Displays OpenFlow group table information. Groups are supported in software tables only. Up to 4 types of groups are supported and 1024 groups across all instances. A select group uses the round-robin method for every packet and the number of action buckets are capped to 8 per group. Groups can be filtered based on group-ID. `show openflow instance <instance-name> groups <group-id>`

Example

```

show openflow instance test groups
Group ID          : 1
Group Type        : ALL
Reference Count    : 32767

```

```

Packet Count      : 0
Byte Count        : 0
Duration          : 10 seconds
Action Buckets    : 1, 2
  Bucket 1
    Packet Count   : 0
    Byte Count     : 0
    Watch port     : Any
    Weight         : 0
    Actions        : output A1
  Bucket 2
    Actions        :output F2
    Packet Count   : 0
    Byte Count     : 0
    Watch port     : Any
    Weight         : 0
    Action         : Output F23

Group ID          : 1
Group Type        : SELECT
Reference Count   : 0
Packet Count      : 0
Byte Count        : 0
Duration          : 10
Action Buckets    : 1
  Bucket 1
    Packet Count   : 0
    Byte Count     : 0
    Watch Port     : Any
    Weight         : 1
    Actions        : output A

Group ID          : 7
Group Type        : INDIRECT
Reference Count   : 0
Packet Count      : 0
Byte Count        : 0
Duration          : 10
Action Buckets    : 1
  Bucket 1
    Packet Count   : 0
    Byte Count     : 0
    Watch Port     : Any
    Weight         : 0
    Actions        : output A1
Group ID          : 32
Group Type        : FAST
  FAIL OVER
Reference Count   : 0
Packet Count      : 0
Byte Count        : 0
Duration          : 10
Action Buckets    : 1
  Bucket 1
    Packet Count   : 0
    Byte Count     : 0
    Watch port     : A1
    Weight         : 0
    Actions        : output A1

```

Example

```
switch(config)# show openflow instance test groups
```

OpenFlow Instance Groups

```
Group ID           : 2
Group Type          : All
Group Location      : Hardware
Is Hardware Counter Available : No
Reference Count     : 1
Packet Count        : 0
Byte Count          : 0
Duration            : 30
Action Buckets      : 1, 2
Bucket 1
  Packet Count      : 0
  Byte Count        : 0
  Watch port        : Any
  Weight            : 0
  Action            : Output F22
Bucket 2
  Packet Count      : 0
  Byte Count        : 0
  Watch port        : Any
  Weight            : 0
  Action            : Output F23
```

Rule pointing to above Group:

Flow 3

Match

```
Incoming Port      : Any Ethernet Type      : IP
Source MAC          : Any Destination MAC    : Any
Source MAC Mask     : 000000-000000
Destination MAC Mask : 000000-000000
VLAN ID             : Any VLAN priority      : Any
Source IP Address    : 11.11.11.11/32
Destination IP Address : Any
IP Protocol          : Any
IP ECN               : Any IP DSCP           : Any
Source Port          : Any Destination Port  : Any
```

Attributes

```
Priority            : 32768      Duration            : 36 seconds
Hard Timeout        : 0 seconds  Idle Timeout      : 0 seconds
Byte Count          : NA         Packet Count      : 0
Flow Table ID       : 100        Controller ID     : listen-port
Cookie              : 0x0
Hardware Index: 17
```

Instructions

```
Apply Actions
  Group             : 2
```

Show per flow rate limiter information

Displays per-flow rate limiters information.

Syntax

```
show openflow instance <instance-name> limiters
```

Example: Display per-flow rate-limiters for an OpenFlow instance

```
switch# show openflow instance test limiters
```

```
OpenFlow Instance Per Flow Rate Limiters
Maximum Limiters   :    256
```

Limiters ID	Action	Rate (kbps)	Flow Count
112	Drop	128	2

Viewing message statistics for an instance

This command displays statistics for flow, port, group, meter modification, and table-features message from the controller, the number of messages of that type received from the controller and the number of messages rejected.

Syntax

```
show openflow instance <instance-name> message-statistics
```

Show message statistics information for an instance. This command displays the number of OpenFlow modification messages received from the controller and the number of messages rejected by the switch.

Example: Display message-statistics for an OpenFlow instance

```
switch(of-inst-test)# show openflow instance test message-statistics
OpenFlow
Message Type                                Received    Rejected
-----
OFPT_FLOW_MOD                             11          0
OFPT_PORT_MOD                             0           0
OFPT_GROUP_MOD                             0           0
OFPT_METER_MOD                             1           0
OFPMP_TABLE_FEATURES (Change Pipeline)    0           0
OFPMP_TABLE_FEATURES (View Pipeline)      1           0
```

Show meter information for a specific instance

Displays meter information. Meters are instance-specific. Meters are supported only in hardware tables; the maximum number of meters differs between platforms. DSCP remark meter is supported only in standard match mode. Further, DSCP remark type band meter cannot be attached to flows with a non-IP match.



NOTE: When a same OpenFlow meter is used two or more times in an OpenFlow pipeline, it results in the meter rates being skewed, leading to unpredictable behavior in how the packets are metered. HPE recommends not using a meter more than once in a packet pipeline to avoid undesired behaviors.

Syntax

```
show openflow instance <instance-name>
```

Example: Display meters for an OpenFlow instance

```
switch(of-inst-test)# show openflow instance test meters
OpenFlow Instance Meters
Meter ID           : 17
```

Flow Count	:	0
Input Packet Count	:	0
Input Byte Count	:	0
Duration	:	6
Band Type	Rate	Packet/Byte Count
-----	-----	-----
Drop	500 kbps	0

Viewing port statistics per instance

Syntax

```
show openflow instance <instance-name> port-statistics
```

Displays port statistics information per instance.

Example: Display port statistics for version 1.3

```
switch# show openflow instance test port-statistics
Number of Ports : 2
Port 47 : Up
Status
Admin. Status : Enabled Flood : Enabled
Receive : Enabled Forward : Enabled
Packet_in : Enabled
Statistics
Collisions : 0
Rx Packets : 0 Tx Packets : 68
Rx Bytes : 0 Tx Bytes : 8066
Rx Dropped : 0 Tx Dropped : 0
Rx Errors : 0 Tx Errors : 0
Frame Errors : 0
CRC Errors : 0
Overrun Errors : 0
Port 48: Down
Status
Admin. Status : Flood :
Receive : Forward :
Packet_in :
Statistics
Collisions :
Rx Packets : 0 Tx Packets : 0
Rx Bytes : 0 Tx Bytes : 0
Rx Dropped : 0 Tx Dropped : 0
Rx Errors : 0 Tx Errors : 0
Frame Errors : 0
CRC Errors : 0
Overrun Errors : 0
```

Diagnostic Tools Overview and Usage

Debug OpenFlow

You can display OpenFlow protocol packets or event description.



NOTE: The `debug openflow packets` option displays only OpenFlow protocol packets exchanged between the switch and the controller.

Syntax

```
switch# debug openflow <errors|events|instance|packets>
```

errors

Display OpenFlow error messages.

events

Enable debug messages for all OpenFlow events like addition/deletion/modification, enable/disable.

instance

Specify an OpenFlow instance for instance-specific debug messages.

packets

Enable debug messages for all OpenFlow packets.

Example: Debug logs

Flow deletion

```
mOFCtrlTask: 00020| DBG|Flow deletion:
idle_timeout=60,d1_type=0x0800,in_port=27,d1_vlan=65535,
d1_vlan_pcp=0,d1_src=00:50:56:9f:5f:0a,d1_dst=00:50:56:9f:19:92,
nw_src=1.2.3.6,nw_dst=1.2.3.4,icmp_type=0,icmp_code=0,
actions=output:26
```

Flow addition

```
mOFCtrlTask: 00019| DBG|Flow addition:
idle_timeout=60,d1_type=0x0800,in_port=27,
d1_vlan=65535,d1_vlan_pcp=0,d1_src=00:50:56:9f:5f:0a,
d1_dst=00:50:56:9f:19:92,nw_src=1.2.3.6,nw_dst=1.2.3.4,
icmp_type=0,icmp_code=0,actions=output:26
```

Flow expiry

```
mOFCtrlTask: 00018| DBG|Flow expiry:
idle_timeout=1200,d1_type=0x0800,nw_src=1.2.3.7,
nw_dst=1.2.3.8,actions=mod_nw_src:9.8.7.6
```

Error messages

Interoperability error messages

Enabling OpenFlow

Enabling OpenFlow when Meshing is enabled results in an error message similar to the following: OpenFlow cannot be enabled when Meshing is configured.

Enabling meshing

Enabling meshing when OpenFlow is enabled results in an error message similar to the following: Meshing cannot be configured when OpenFlow is enabled.

Enable OpenFlow with QinQ

Enabling OpenFlow when Q-in-Q is enabled results in an error message similar to the following: OpenFlow cannot be enabled when Q-in-Q is configured.

Enabling QinQ with OpenFlow

Enabling Q-in-Q when OpenFlow is enabled results in an error message similar to the following: Q-in-Q cannot be configured when OpenFlow is enabled.

Enabling transparent mode

Enabling Transparent Mode (TRmode) when OpenFlow is enabled results in an error message similar to the following: Transparent Mode cannot be enabled when OpenFlow is enabled.

Enabling OpenFlow with transparent mode

Enabling OpenFlow when Transparent Mode is enabled results in an error message similar to the following: OpenFlow cannot be enabled when Transparent Mode is enabled.

Enabling remote mirror endpoint

Enabling Remote Mirror Endpoint when OpenFlow is enabled generates an error message similar to the following: Remote Mirror Endpoint cannot be configured when OpenFlow is enabled.

Enabling OpenFlow with remote mirror endpoint

Enabling OpenFlow when Remote Mirror Endpoint is enabled generates an error message similar to the following: OpenFlow cannot be enabled when Remote Mirror Endpoint is configured.

Adding a port

Adding a port to a trunk that is part of an OpenFlow member VLAN generates an error message similar to the following.

Trunk in use by an OpenFlow instance may not be modified.

Deleting a port

Deleting a port from a trunk that is part of an OpenFlow member VLAN generates an error message similar to the following.

Trunk in use by an OpenFlow instance may not be modified.

Moving a trunk

When moving a trunk that is part of an OpenFlow member VLAN from one VLAN to another, the VLAN generates an error message similar to the following:

Trunk in use by an OpenFlow instance may not be moved.

Tagging/Untagging trunk

Toggling membership of the trunk from tagged to untagged when that trunk is part of an OpenFlow member VLAN generates an error message similar to the following.

Trunk in use by an OpenFlow instance may not be modified.

Enable LACP

Trying to enable LACP while OpenFlow is enabled generates the following error message.

LACP cannot be configured when OpenFlow is enabled.

Enable OpenFlow

Trying to enable OpenFlow when LACP is enabled generates the following error message.

OpenFlow cannot be configured when LACP is enabled.

Show per-flow rate limiters

Trying to show per-flow rate limiters for an instance running OpenFlow version 1.3 generates an error message similar to the following.

This command is supported only for an OpenFlow version 1.0 instance.

no allow-V1-module

Trying to run the command `no allow-V1-module` when OpenFlow is enabled, generates an error message similar to the following.

V1 modules cannot be disabled when OpenFlow is enabled.

allow-V1-module

Trying to run the command `allow-V1-module` when OpenFlow is enabled generates an error message similar to the following.

V1 modules cannot be enabled when IP Control Table Mode is enabled.

Non-compatible mode

Trying to enable OpenFlow when a switch is in a non-compatible mode (no allow-V1-module) generates an error message similar to the following.

OpenFlow cannot be enabled when V1 modules are disabled.

Enable virus throttling

Trying to enable virus throttling when OpenFlow is enabled generates an error message similar to the following.

Virus throttling cannot be enabled when OpenFlow is enabled.

Enable OpenFlow with virus throttling

Trying to enable OpenFlow when virus throttling is enabled generates an error message similar to the following.

OpenFlow cannot be configured when virus throttling is enabled.

Controller error messages

Deleting an unconfigured controller

Attempt to delete a controller that has not been configured results in an error message similar to the following: `switch(vlan-3)# no openflow controller-id 2 [controller-id] 2 not found.`

Configure or modifying an existing controller

Attempting to configure a controller that exists or modifying the parameters of an existing controller results in an error message similar to the following: `A controller is already configured with this ID.`

Associated controllers

Attempting to delete existing controllers previously associated with an OpenFlow instance result in an error message similar to the following: Controller cannot be removed when in use by an OpenFlow instance.

Setting IP Control Table mode

Attempting to set IP Control Table Mode when the switch is in compatible mode results in an error message similar to the following:

```
IP Control Table Mode cannot be set when V1 module is enabled.
```

Specifying an invalid flow table

Attempting to specify an invalid flow table ID results in an error message similar to the following:

```
Invalid flow table number
```

Listen port or controller error

Commands issued from listen port or controllers are not successful.

Procedure

1. Enable `debug openflow`, which displays the switch output, helping you determine whether the error occurs at the switch or the controller.
2. Enable `debug openflow instance [instance-name]` to further identify the error.
3. Verify the packet capture for the request and reply to isolate whether the error is occurring at the switch or the controller.



NOTE: This problem occurs if some controllers do not fully conform to the specification and therefore cannot handle replies from the switch. The replies in the packet capture are visible from the switch, but not from the controller.

4. Enable `debug destination session` to further identify the error.

Specifying a port

An attempt to specify an application port that is out of range results in an error message similar to the following:

```
Invalid port. Valid range is 1024-65534.
```

Port error messages

Egress-only ports

Trying to enable or disable egress-only ports when OpenFlow is enabled generates an error message similar to the following.

```
Egress only ports can be set only when OpenFlow is disabled.
```

Limiter error messages

Removing limiters

Trying to remove a limiter when none are configured for an instance generates an error message similar to the following.

```
No limiters found for this OpenFlow instance.
```

VLAN error messages

Member to controller VLAN

Specifying a member VLAN as a controller VLAN results in an error message similar to the following: The specified VLAN is already member of OpenFlow instance *instance-name* and hence cannot be added as controller interface.

VLAN in an OpenFlow instance

Specifying a VLAN that is already a part of a different OpenFlow instance results in an error message similar to the following: The VLAN specified is already a member of another OpenFlow instance.

VLAN range

Specifying a VLAN that is outside the allowed VLAN range results in an error message similar to the following: Invalid Input : *VLAN-ID*

Management VLAN

When the user tries to add the management VLAN to an OpenFlow instance results in an error message similar to the following: A management VLAN cannot be a member of an OpenFlow instance.

Configure VLAN as management

When the user tries to configure an OpenFlow instance VLAN as management VLAN results in an error message similar to the following: Management VLAN cannot be configured. VLAN *<n>* is member of an OpenFlow instance.

Dynamic VLAN

When a dynamic VLAN is added as a member VLAN, the result is an error message similar to the following: Dynamic VLAN cannot be added as a member VLAN.

Controller interface

Adding a controller interface as member VLAN results in an error message similar to the following: Controller interface cannot be added as member VLAN.

Instance error messages

Enable a named instance

Attempting to enable a named instance without a listen port or controller and a member VLAN displays an error message similar to the following: A controller and a member VLAN must be added to the named instance before enabling it.

Enable an aggregate instance

Attempting to enable an aggregate instance without a listen port or controller displays an error message similar to the following: A listen-port or a controller must be added to the aggregate instance before enabling it.

Maximum number of instances

Configuring an instance when the maximum number of OpenFlow instances is already configured displays an error message similar to the following: Maximum number of OpenFlow instances (128) already configured.

Instance name that exceeds length

Configuring an instance with a name that exceeds the maximum length requirement displays an error message similar to the following: Maximum length of the instance-name is 32 characters.

Create an aggregate instance

Trying to create an aggregate instance when a named instance exists on the switch displays an error message similar to the following: An aggregate instance cannot be created when named instances exist.

Create a named instance

Trying to create a named instance when an aggregate instance is already configured displays an error message similar to the following: Named instances cannot be created when an aggregate instance exists.

Deleting an instance

Trying to delete a nonexistent instance displays an error message similar to the following: Instance not found.

Enabling an instance

Attempting to enable an OpenFlow instance without configuring a listen port or a controller displays an error message similar to the following: A listen-port or a controller, and a member VLAN must be added to the named instance before enabling it.

Delete a member

Trying to delete a member that does not belong to the instance displays an error message similar to the following: VLAN *VLAN-ID* is not a member of this instance.

Modifying backoff interval

Trying to modify the backoff interval when the instance is enabled displays an error message similar to the following: Instance configuration cannot be modified when the instance is enabled.

Instance name

When naming an instance, only alphabetic characters, numerals, and underscores are allowed in the instance name. Failure to follow this rule results in an error message similar to the following: Invalid name. Only alphanumeric characters and underscores are allowed.

Errors concerning auxiliary connections

Removing auxiliary connection

Removing an auxiliary connection that is associated displays an error message similar to the following: Auxiliary connection is in use by an OpenFlow instance and cannot be removed.

Deleting unconfigured auxiliary connection

Deleting an auxiliary connection that has not been configured displays an error message similar to the following: Auxiliary connection *index* not found.

Associating multiple auxiliary indexes

Associating more than one auxiliary index displays an error message similar to the following. Only one auxiliary connection can be configured per main controller connection.

Associating unconfigured auxiliary indexes

Associating an auxiliary index that is not configured, displays an error message similar to the following. No auxiliary connection is configured with this index.

Checking for the static limit

Checking for the static limit and error out while configuring displays an error message similar to the following. Maximum number of auxiliary connections configured.

Associating auxiliary connection

Associating an auxiliary connection to an instance running version 1.0 displays an error message similar to the following. Auxiliary connection can only be associated with instance running version 1.3 and above.

Associating multiple auxiliary connections

Associating more than one auxiliary connection to an instance controller connection displays an error message similar to the following. Only one auxiliary connection can be configured per main controller connection.

Other scenarios

Setting policy engine resource usage when OpenFlow is enabled

When the policy engine resource usage is set while OpenFlow is enabled, an error message similar to the following is displayed: Resource usage can be set only when OpenFlow is disabled.

Securing a connection with no certificate configured

When securing a connection with no certificate configured for OpenFlow, an error message similar to the following is displayed: Certificate for OpenFlow is not configured.

Setting the protocol version with instances enabled

Setting the protocol version when instances are enabled results in an error message similar to the following: Instance configuration cannot be modified when the instance is enabled.

Troubleshooting an instance

To troubleshoot instances, check the following.

- To connect a controller, there must be ip-connectivity between controller and switch over the controller VLAN.
- The controller must be capable of negotiating to a version equal to or less than the configured or supported version
- Use the `show openflow` command to check instances.

Example

```
switch(openflow)# show openflow
OpenFlow : Enabled
Egress Only Ports Mode : Disabled
Instance Information
No. of No. of OpenFlow
Instance Name Oper. Status H/W Flows S/W Flows Version
-----
test Up 1 4 1.0
```

Use the `show openflow instance <instance-name>` command.

Example

```
switch(openflow)# show openflow instance test
Configured OF Version : 1.0
Negotiated OF Version : 1.0
Instance Name : test
Data-path Description : test
Administrator Status : Enabled
Member List : VLAN 3
Pipeline Model : Standard MatchListen Port : 6633
Operational Status : Up
Operational Status Reason : NA
Datapath ID : 000340a8f09e8600
Mode : Active
Flow Location : Hardware and Software
```

```

No. of Hardware Flows : 1
No. of Software Flows : 4
Hardware Rate Limit : 0 kbps
Software Rate Limit : 100 pps
Conn. Interrupt Mode : Fail-Secure
Maximum Backoff Interval : 60 seconds
Probe Interval : 10 seconds
Hardware Table Miss Count : 0
No. of Software Flow Tables : NA
Egress Only Ports : None
Table Model : Single Table
Source MAC Group Table : Disabled
Destination MAC Group Table : Disabled
Controller Id Connection Status Connection State Secure Role
-----
1 Connected Active No Equal

```

Commands issued from listen port or controllers are not successful

When commands issued from the listen port or the controller are not successful, the following commands can be used to isolate and troubleshoot the problem.

Syntax

```
show openflow instance <instance-name> flow-table <flow-table-number> table-capabilities
```

To display the table-capabilities of the instance in OpenFlow 1.3.

Syntax

```
show vlan <member-vlan-no>
```

To display the port-description.

Syntax

```
show openflow instance <instance-name>
```

To see egress-only ports.

Syntax

```
show openflow instance <instance-name> port-statistics
```

To display port statistics.

Syntax

```
show openflow instance <instance-name> meters <meter-id>
```

To display the meter statistics.

Syntax

```
show openflow instance <instance-name> groups
```

To display groups.

Connection interruption mode setting

For minimal impact to an underlying network when a switch loses connection to the controller, the recommended setting is fail-standalone mode.

Flow modification

Add/Modify/Delete flow

When the switch rejects a request to add, modify, or delete a flow mod, use the following command:

Syntax

```
show openflow instance <instance-name> message-statistics
```

Example

```
switch(of-inst-test)# show openflow instance test message-statistics
```

OpenFlow Message Type	Received	Rejected
-----	-----	-----
OFPT_FLOW_MOD	11	0
OFPT_PORT_MOD	0	0
OFPT_GROUP_MOD	0	0
OFPT_METER_MOD	0	0
OFPMPT_TABLE_FEATURES (Change Pipeline)	0	0
OFPMPT_TABLE_FEATURES (View Pipeline)	1	0

Verifying flows

The flow can be verified at the switch by using the following command.

Syntax

```
show openflow instance <instance-name> flows
```

Enable debug `openflow` at the switch. Run the command and observe the debug output for more specific reasons why the switch rejected the flow.



NOTE: Similar troubleshooting techniques can be employed for port-modification, meter-modification, and group-modification issues.

Programming flow errors

When programming flows via a controller, error messages may be returned based on implementation restrictions in the OpenFlow switch. Examples relevant to OpenFlow 1.3 include:

- Table 0 restrictions
- Table 0, a read-only table, in the OpenFlow 1.3 multiple pipeline represents the start of the pipeline.

IP control table restrictions

The following may have caused the error conditions for Table 50:

- Table-miss rule is read-only.
- Only unicast IP addresses can be used as match parameters in a flow.
- Only "Goto" instruction is supported by this table.
- Flow with invalid VLAN match parameter is not allowed. A VLAN that does not exist on the switch is considered invalid.

Possible errors returned to the controller:

OFPERR_OFPMFC_EPERM

OFPERR_OFPMFC_TABLE_FULL

OFPERR_OFPBMC_BAD_FIELD (Bad or unsupported match parameter in the flow)

OFPERR_OFPBAC_BAD_TYPE (Bad or unsupported action in the flow)

OFPERR_OFPBIC_BAD_TABLE_ID

OFPERR_OFPMFC_UNKNOWN (Any internal system error)

Policy engine table restrictions

Error conditions for Table 100, 101, or 102 may result from the following:

- In Aggregate mode, an Output-Port action is allowed only if the flow has VLAN as a match field or has as a Modify-VLAN action specified.
- Modify VLAN-PCP and P-ToS are the only Set-field actions allowed along with Output:NORMAL action.

OpenFlow V1.0 instance

OpenFlow 1.0 exposes a single table to the controller. The action of the default table-miss rule for such an instance is "Goto Controller".

OpenFlow 1.3 instance

OpenFlow 1.3 instance exposes a multi-table model. For every table, the action of the default table-miss rule is "DROP". The controller must appropriately modify the table-miss rule for every table, for traffic to traverse the multi-table pipeline.

Example

Consider the table model in standard mode comprised of tables 0, 100, 200-201-202-203:

- If a rule is programmed into table 200 that permits traffic, then unless the default "table-miss rule" for table 100 is modified, traffic does not pass to table 200.
- If the default table-miss rule of table 100 is modified with the action given as "Goto table 200", traffic proceeds to table 200. A table-miss rule must be programmed similarly for table 200, and so on.



NOTE:

A table-miss rule is a flow with priority of 0 and all match fields wild-carded.

Flows go missing after addition

Verify the idle-timeout/hard-timeout of the flow.

- Table 50 supports only 12 seconds as the minimum hardware refresh rate. Flows programmed in this table must have at least an idle-timeout of twice that, or 24 seconds.
- Tables 100/101/102 support a configurable hardware refresh rate. If the "policy-engine-table refresh-interval" is configured for 5 seconds, minimum idle-timeout supported is 10 seconds (double the time configured.)

Missing line rate performance after flows are successfully installed

If an instance is running OpenFlow v1.0 and the flow cannot be accommodated in the hardware or a higher priority overlapping rule is present in software, or we have reached the policy engine usage limit configured, the flow will be housed in the software table. Use the `show openflow instance <instance-name> flows` command to verify that the flow is housed in the software table.

If the flow is programmed in software, line rate performance is not seen in packet forwarding.

To know which flows are accommodated in hardware, see the Flow classification section of this document.

For an OpenFlow1.3 instance, there could be several software tables, 200 to 203.

Troubleshooting scenarios and error messages

How to troubleshoot if instance is not coming up

When an instance is not coming up, use the following commands to troubleshoot the instance status.

Procedure

1. Run the `show openflow` command.

```
switch(openflow)# show openflow

OpenFlow                               : Enabled
Egress Only Ports Mode                 : Disabled

Instance Information

Instance Name      Oper. Status  No. of  No. of  OpenFlow
-----          -
test              Up           0       0       1.0
```

2. Run the `show openflow instance <instance-name>` command.

```
switch(openflow)# show openflow instance test

Configured OF Version      : 1.0
Negotiated OF Version      : 1.0
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model             : Standard Match
Listen Port                : 6633
Operational Status         : Up
Operational Status Reason  : NA
Datapath ID                : 000340a8f09e8600
Mode                       : Active
Flow Location              : Hardware and Software
No. of Hardware Flows      : 0
No. of Software Flows      : 0
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps
Conn. Interrupt Mode       : Fail-Secure
Maximum Backoff Interval   : 60 seconds
Probe Interval             : 10 seconds
```



```
Hardware Table Miss Count      : 0
No. of Software Flow Tables   : NA
Egress Only Ports             : None
Table Model                   : Single Table
Source MAC Group Table        : Disabled
Destination MAC Group Table    : Disabled
```

Controller Id	Connection Status	Connection State	Secure	Role
1	Disconnected	Backoff	No	Equal

Reporting problems

If you are unable to solve a problem with OpenFlow, do the following:

1. Read the release notes for OpenFlow to see if the problem is known. If it is, follow the solution offered to solve the problem.
2. Determine whether the product is still under warranty or whether your company purchased support services for the product. Your operations manager can supply you with the necessary information.
3. If the problem you are experiencing has already been reported, access **HPE Support Center** and search the technical knowledge databases to determine the type of documentation and resources you have access to, which depends on your level of entitlement.



NOTE: The HP Support Center at **HPE Support Center** offers peer-to-peer support to solve problems and is free to users after registration.

If this problem is new or if you need additional help, log your problem with the HP Support Center, either on line through the support case manager at **HPE Support Center**, or by calling HPE Support. If your warranty has expired or if you do not have a valid support contract for your product, you can still obtain support services for a fee, based on the amount of time and material required to solve your problem.

4. If you are requested to supply any information pertaining to the problem, gather the necessary information and submit it. The following sections describe some of the information that you may be asked to submit.

OpenFlow Flow-Mod and Pipeline-Mod error message enhancements

Users can track the reasons why a flow or a pipeline is being rejected by a switch.

Operation notes

- More descriptive reasoning is given with specific OpenFlow error codes which helps explain why the switch rejected a particular flow or pipeline.
- Supported only for instances running OpenFlow 1.3.
- Supported on v1, v2, and v3 of ASICs.
- Supported on all pipeline-models — standard-match, ip-control, and custom.
- Statistics are tracked per OpenFlow instance.

Restrictions

Not all OpenFlow flow-mod and pipeline-mod related failures are tracked.

Networking Websites

Aruba Support Portal

asp.arubanetworks.com

Aruba Software and Documentation

asp.arubanetworks.com/downloads

Hewlett Packard Enterprise Networking Software

www.hpe.com/networking/software

Hewlett Packard Enterprise Networking website

www.hpe.com/info/networking

Hewlett Packard Enterprise My Networking website

www.hpe.com/networking/support

Hewlett Packard Enterprise My Networking Portal

www.hpe.com/networking/mynetworking

Hewlett Packard Enterprise Networking Warranty

www.hpe.com/networking/warranty

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

For additional websites, see [Support and other resources](#).

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:
<http://www.hpe.com/info/assistance>
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:
<http://www.hpe.com/support/hpesc>

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

Hewlett Packard Enterprise Support Center

www.hpe.com/support/hpesc

Hewlett Packard Enterprise Support Center: Software downloads

www.hpe.com/support/downloads

Software Depot

www.hpe.com/support/softwaredepot

- To subscribe to eNewsletters and alerts:
www.hpe.com/support/e-updates
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:
www.hpe.com/support/AccessToSupportMaterials



IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

<http://www.hpe.com/support/selfrepair>

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Datacenter Care services

www.hpe.com/services/datacentercare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

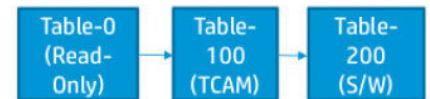
Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hpe.com). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

Hardware differences between v1, v2, and v3 modules affect flow match and capabilities. For additional information about modules, see the latest Release Notes for your switch.

Hardware match chart

Standard Match Model (v1) K_15_17



OpenFlow Match (H/W) [1.5k Rules on V1]

The following OpenFlow fields can be matched in hardware

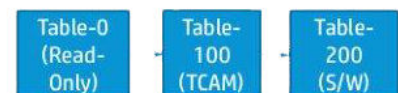
IN PORT	VLAN ID	ETHERNET TYPE	IP SRC (v4/v6)	IP DST (v4/v6)	IP PROTO	IP DSCP	TCP/UDP SRC	TCP/UDP DST	TCP Flags (Experimenter Match)	L4 port Ranges (Experimenter Match)

OpenFlow Action (H/W)

The following OpenFlow actions can be executed in hardware

O/P Single Port	Modify VLAN PRIORITY	Modify IP DSCP

Standard Match Model (v2 & v3) K, KA, KB, W_15_17



OpenFlow Match (H/W) [4k Rules on V2/V3, 1k Rules on 2920]

The following OpenFlow fields can be matched in hardware

IN PORT	VLAN ID	VLAN PRIORITY	MAC SA	MAC DA	ETHERNET TYPE	IP SRC (v4/v6)	IP DST (v4/v6)	IP PROTO	IP DSCP	TCP/UDP SRC	TCP/UDP DST

TCP FLAGS (EXPERIMENTER MATCH)	L4 PORT RANGES (EXPERIMENTER MATCH)

OpenFlow Action (H/W)

The following OpenFlow actions can be executed in hardware

O/P One or More Ports	Modify MAC SA	Modify MAC DA	Modify VLAN ID	Modify VLAN PRIORITY	Modify IP DSCP

SUPPORTED 1.3v ONLY

Custom Pipeline Model (v3)

KB_15_17

OpenFlow Match (H/W) [Scale discussed in subsequent slides]

The following OpenFlow fields can be matched in hardware

IN PORT	VLAN ID	VLAN PRIORITY	MAC SA	MAC DA	ETHERNET TYPE	IP SRC (v4/v6)	IP DST (v4/v6)	IP PROTO	IP DSCP	TCP/UDP SRC	TCP/UDP DST
---------	---------	---------------	--------	--------	---------------	----------------	----------------	----------	---------	-------------	-------------

OpenFlow Action (H/W)

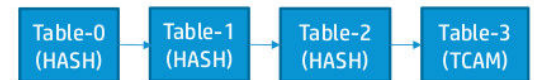
The following OpenFlow actions can be executed in hardware

O/P One or More Ports	Modify MAC SA	Modify MAC DA	Modify VLAN ID	Modify VLAN PRIORITY	Modify IP DSCP	Modify IPv4 SRC	Modify IPv4 DST	Modify TCP/UDP SRC (IPv4)	Modify TCP/UDP DST (IPv4)
-----------------------	---------------	---------------	----------------	----------------------	----------------	-----------------	-----------------	----------------------------	---------------------------

Custom Pipeline Model - Default Pipeline

- All the OpenFlow Tables in this mode are in **hardware**.

Table-Number/ Property	Table-0 (HASH)	Table-1 (HASH)	Table-2 (HASH)	Table-3 (TCAM)
Match Key	VLAN_VID & ETH_SRC	VLAN_VID & ETH_DST	ETH_TYPE, IP_SRC, IP_DST, IP_PROTO, SRC_PORT & DST_PORT	IN_PORT, ETH_SRC, ETH_DST, VLAN_VID, VLAN_PCP, ETH_TYPE, IP_SRC, IP_DST, IP_DSCP, IP_PROTO, SRC_PORT & DST_PORT
Instructions	APPLY, WRITE, CLEAR, METER, GOTO	APPLY, WRITE, CLEAR, METER, GOTO	APPLY, WRITE, CLEAR, METER, GOTO	APPLY, WRITE, CLEAR, METER,
Scale	8k	8k	8k	2k



And what's more - The pipeline can be changed on the fly by the controller using standard OpenFlow constructs (OFPMP_TABLE_FEATURES message). **Create a network pipeline of your choice!**

Custom Pipeline - SCALE

The scale is dependent on the Type and Match Fields of the customized tables

▪ HASH TABLE (No Wildcards allowed)

- ☐ In a single hash table we can match fields from any 3 Groups out of the 4.
- ☐ Based on the number of groups matched, scale will vary
 - ☐ One Group → Up to 64K entries
 - ☐ Two or Three Groups → Up to 32K entries.

▪ TCAM TABLE (Wildcards allowed)

- ☐ In a single TCAM table we can match fields from all 4 Groups.
- ☐ Based on the number of groups matched, scale will vary:
 - ☐ One Group → Up to 8K entries
 - ☐ Two Groups → Up to 4K entries
 - ☐ Three or Four Groups → Up to 2K entries

Group 1	IN PORT	MAC SA	MAC DA	ETHER TYPE		
Group 2	IP SA (IPv4 and IPv6)					
Group 3	VLAN NUM	VLAN PCP	IP DSCP	IP PROTO	TCP/UDP SRC Port	TCP/UDP DST Port
Group 4	IP DA (IPv4 and IPv6)					
Group 5	VLAN UNTAGGED	VLAN TAGGED	VLAN ID			
Group 6	ARP OP					

OpenFlow 1.3 multi-table model and device modes

Figure 16: *Standard mode (default)*

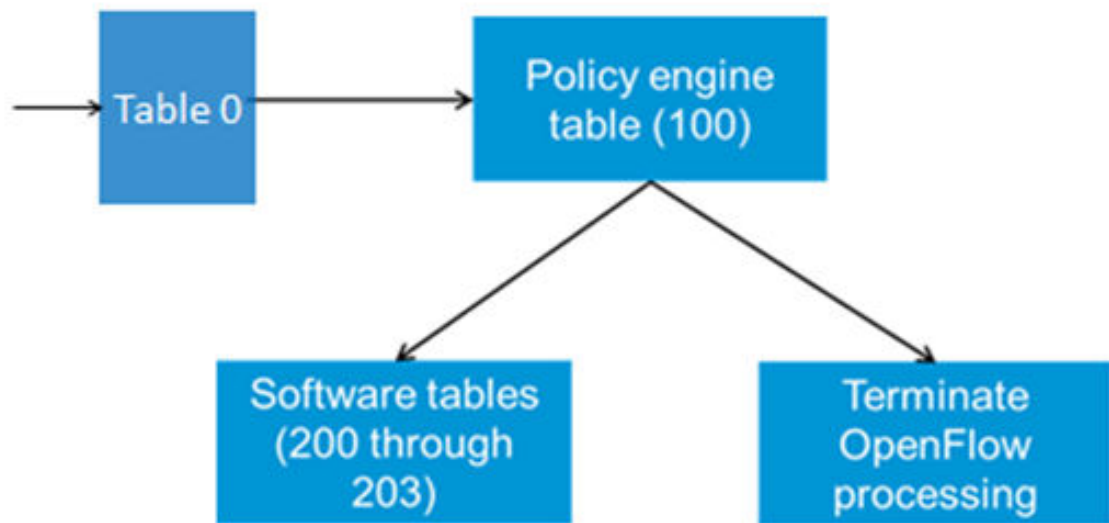


Figure 17: *IP control table default*



Table 7: Device modes and OpenFlow table model

OpenFlow Protocol Version	Switch Mode	Table Model	Number of tables	Matching ability	Actions in Hardware (Nov 2013)
v1.0	Compatible Mode – “allow-v1-modules” (v1 and v2) acts as v1	Single table only	1	Same as 15_10	Same as 15_10
v1.0	Non Compatible Mode – “no allow-v1-modules” (v2 only) act as v2	Single table only	1	Full 12 tuple match in policy engine	Same as 15_10 Plus new actions – rewrite VLAN ID, rewrite MAC address, forward to multiple ports
v1.3	Compatible Mode (v1 and v2) acts as v1	Standard Match	1 policy engine + 1 software	Same as 15_10	Same as 15_10
v1.3	Non-Compatible Mode (v2 only) act as v2	Standard Match	1 policy engine + 1 software	Full 12 tuple match in policy engine	Same as 15_10 Plus new actions - rewrite VLAN ID, rewrite MAC address, forward to multiple ports
v1.3	Non-Compatible Mode (v2 only) act as v2	IP Control Mode	1 IP Control Table + 1 Policy Engine Table + 1 software Table.	{src. VLAN, src. IP, dst. IP} in IP control table + Full 12 tuple match in policy engine	Same as 15_10 Plus new actions in - rewrite VLAN ID, rewrite MAC address, forward to multiple ports

Flow table capabilities

Match/Set-Field

OpenFlow instance running in custom pipeline-model supports tables with the following base class match fields (not all in a single table).

Field	Match Supported	Mask Supported	Supported since	Set-Field Supported	Supported since
OFPXMT_OFB_ETH_DST	Yes	Yes	15.17	Yes	15.17
OFPXMT_OFB_METADATA	Yes	Yes	15.17	Yes	15.17
OFPXMT_OFB_VLAN_VID	Yes	No	15.17	Yes	15.17
OFPXMT_OFB_VLAN_PCP	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_IP_DSCP	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_IPV4_SRC	Yes	Yes	15.17	Yes	15.17
OFPXMT_OFB_IPV4_DST	Yes	Yes	15.17	Yes	15.17
OFPXMT_OFB_TCP_SRC	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_TCP_DST	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_UDP_SRC	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_UDP_DST	Yes	N/A	15.17	Yes	15.17
OFPXMT_OFB_IN_PORT	Yes	N/A	15.17	N/A	N/A
OFPXMT_OFB_IN_PHY_PORT	N/A	N/A	N/A	N/A	N/A
OFPXMT_OFB_METADATA	No	No	N/A	N/A	N/A
OFPXMT_OFB_ETH_TYPE	Yes	N/A	15.17	N/A	N/A
OFPXMT_OFB_IP_ECN	No	N/A	N/A	N/A	N/A
OFPXMT_OFB_IP_PROTO	Yes	N/A	15.17	N/A	N/A
OFPXMT_OFB_SCTP_SRC	No	N/A	N/A	No	N/A
OFPXMT_OFB_SCTP_DST	No	N/A	N/A	No	N/A
OFPXMT_OFB_ICMPV4_TYPE	Yes	N/A	16.02	No	N/A
OFPXMT_OFB_ICMPV4_CODE	Yes	N/A	16.02	No	N/A
OFPXMT_OFB_ARP_OP	Yes	N/A	16.02	No	N/A

Table Continued

Field	Match Supported	Mask Supported	Supported since	Set-Field Supported	Supported since
OFPXMT_OFB_ARP_SPA	Yes	Yes	16.02	No	N/A
OFPXMT_OFB_ARP_TPA	Yes	Yes	16.02	No	N/A
OFPXMT_OFB_ARP_SHA	Yes	Yes	16.02	No	N/A
OFPXMT_OFB_ARP_THA	Yes	Yes	16.02	No	N/A
OFPXMT_OFB_IPV6_SRC	Yes	Yes	15.17	No	N/A
OFPXMT_OFB_IPV6_DST	Yes	Yes	15.17	No	N/A
OFPXMT_OFB_IPV6_FLABEL	Yes	Yes	16.02	No	N/A
OFPXMT_OFB_IPV6_ND_SLL	No	N/A	N/A	No	N/A
OFPXMT_OFB_IPV6_ND_TLL	No	N/A	N/A	No	N/A
OFPXMT_OFB_MPLS_LABEL	No	N/A	N/A	No	N/A
OFPXMT_OFB_MPLS_TC	No	N/A	N/A	No	N/A
OFPXMT_OFB_MPLS_BOS	No	N/A	N/A	No	N/A
OFPXMT_OFB_PBB_ISID	No	No	N/A	No	N/A
OFPXMT_OFB_TUNNEL_ID	No	No	N/A	No	N/A
OFPXMT_OFB_IPV6_EXTHDR	No	No	N/A	No	N/A

Instructions

Instruction	Supported	Supported since
OFPIT_GOTO_TABLE	Yes	15.17
OFPIT_WRITE_METADATA	No	N/A
OFPIT_WRITE_ACTIONS	Yes	15.17
OFPIT_APPLY_ACTIONS	Yes	15.17

Table Continued

Instruction	Supported	Supported since
OFFPIT_CLEAR_ACTIONS	Yes	15.17
OFFPIT_METER	Yes	15.17

Actions

Action	Supported	Supported since
OFFPAT_OUTPUT	Yes	15.17
OFFPAT_COPY_TTL_OUT	No	
OFFPAT_COPY_TTL_IN	No	
OFFPAT_SET_MPLS_TTL	No	
OFFPAT_DEC_MPLS_TTL	No	
OFFPAT_PUSH_VLAN	Yes	15.17
OFFPAT_POP_VLAN	Yes	15.17
OFFPAT_PUSH_MPLS	No	
OFFPAT_POP_MPLS	No	
OFFPAT_SET_QUEUE	No	
OFFPAT_GROUP	Yes	15.17
OFFPAT_SET_NW_TTL	Yes	15.17
OFFPAT_DEC_NW_TTL	No	
OFFPAT_SET_FIELD	Yes	15.17
OFFPAT_PUSH_PBB	No	
OFFPAT_POP_PBB	No	

This section documents some of the behaviors exhibited during the implementation of OpenFlow. These behaviors were exposed during testing and may include unit, conformance, integration, interoperability, stress, and system testing.

A hardware flow with an idle timeout of 10 seconds gets deleted even though packets match the flow within the idle timeout

Problem statement

A hardware rule is programmed with idle timeout as 10 seconds and hard timeout as 0. Packets are pumped at 1000 pps to the switch matching the flow. However, after 10 seconds, the rule gets removed from the switch.

Reason for this behavior

By default the hardware statistics refresh rate (set using `openflow hardware-statistics refresh-rate policy-engine-table <seconds>` and information available through `show openflow`) is 20 seconds. This rate means that the packet count statistics get updated only every 20 seconds. So, when the idle timeout is set to less than 20 seconds, when a check is done for flow statistics after 10 seconds, it is not updated. Hence, the flow is deleted.

Customer Note

The user has the option of reducing or increasing the refresh rate. However, the user must be aware of its implications. An increase in refresh rate leads to deletion of flows, which have an idle timeout less than the configured refresh rate. A decrease in refresh rate leads to over-use of the CPU (because of polling hardware statistics more frequently.)

Controller flows — flow in hardware and processing software

Flows with an action to send matching traffic to controller are installed on hardware. But, the actual traffic forwarding takes place in software, since we must add the required OpenFlow specific headers. Due to this characteristic, the actual forwarding does not take place at the line rate. A sample controller flow looks like:

Example

In this example, any packet that comes on port A1, is forwarded to the controller after adding required OpenFlow packet headers (as the packet is sent as a `packet_in`) to the controller. Since this processing is done on software, we cannot send the incoming traffic at line rate.

```
switch(openflow)# show openflow instance test flows
Flow 1
Match
  Incoming Port      : A1                      Ethernet Type      : Any
  Source MAC         : Any                     Destination MAC    : Any
  Destination MAC Mask : 000000-000000          VLAN Priority      : Any
  VLAN ID            : Any
```

```

Source IP Address      : Any
Destination IP Address : Any
IP Protocol           : Any
Source Port           : Any
Attributes
Priority               : 55000
Hard Timeout          : 0 seconds
Byte Count            : 0
Controller ID         : listen-port
Flow Location         : Hardware
Hardware Index        : 0
Reason Code           : 12
Reason Description    : Rule is in hardware.
Actions
  Controller Port
IP ToS Bits           : Any
Destination Port      : Any
Duration              : 4 seconds
Idle Timeout          : 0 seconds
Packet Count          : 0
Cookie                : 0x0

```

DUT matches and processes incoming untagged packets for VLAN id

For certain flows with a match on the VLAN ID, even untagged packets are matched. This matching occurs on untagged ports only. The existing behavior exists because L2 hardware adds the VLAN id and VLAN priority meta-information irrespective of whether the packet came in tagged or untagged. Flows that can be accelerated into hardware are put into hardware, whereas flows that cannot be accelerated in hardware are put into software. The observed behavior is observed for hardware and software flows.

Events that change the Operational Status of the OpenFlow instance

The `Oper. Status` field indicates the operational status of the instance and can be either up or down. The operational status is down when either the member VLAN of the OpenFlow instance does not exist on the switch or the controller VLAN of the OpenFlow instance does not exist on the switch. In the case when multiple controllers connect over multiple controller VLANs, the operational status is down when none of the controller VLANs exist on the switch. When the member VLAN is down - all ports on the member VLAN are down.

For example, the `show openflow instance <instance-name>` displays all the OpenFlow instance-related information as follows:



NOTE: For purposes of this example, the instance **<test>** has been created.

```

switch(openflow)# show openflow instance test

Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model             : Standard Match
Listen Port                : 6633
Operational Status         : Down
Operational Status Reason  : None of the member VLANs are configured
Datapath ID                : 0000000000000000
Mode                       : Active
Flow Location              : Hardware and Software
No. of Hardware Flows      : 0
No. of Software Flows      : 0
Hardware Rate Limit        : 0 kbps
Software Rate Limit        : 100 pps

```

```

Conn. Interrupt Mode      : Fail-Secure
Maximum Backoff Interval : 60 seconds
Probe Interval           : 10 seconds
Hardware Table Miss Count : NA
No. of Software Flow Tables : 1
Egress Only Ports        : None
Table Model              : Policy Engine and Software
Source MAC Group Table    : Disabled
Destination MAC Group Table : Disabled

```

Controller Id	Connection Status	Connection State	Secure	Role
1	Disconnected	Void	No	Equal

OpenFlow influence on CPU generated packets

In some cases, TCAM rules affect the CPU generated packets. This behavior is consistent with OpenFlow versions 1.0 and 1.3. One example of such a case is when a rule is in place with the `in_port` as a wild card but has an SRC IP address that matches the IP address configured on the switch.

OpenFlow 1.0 supports IP address masking

OpenFlow supports IP subnet mask. Controllers can specify the subnet mask associated with an IP address and sent to the OpenFlow switch. The switch accepts the IP address with the subnet mask and associates any packets coming with the subnet mask with the rule.

For example, the K.15.10. OpenFlow implementation supports the ability to match on IP address and subnet mask when the OpenFlow controller programs such flows. Consider this example where the `ovs-ofctl` utility is used to add a flow that matches on a network source address of 1.1.1.1 with a subnet mask of /24. 10.10.10.1 here is the IP address of the switch that has an OpenFlow listen port open on port 6633.

```

openflow@openflow-ubuntu-08:~# ovs-ofctl add-flow
tcp:10.10.0.1:6633 ip,nw_src=1.1.1.1/24,actions=output:1

```

To verify that this flow has been installed on the switch, we run the `ovs-ofctl` command and verify the output.

```

openflow@openflow-ubuntu-08:~# ovs-ofctl dump-flows tcp:10.10.0.1:6633
NXST_FLOW reply (xid=0x4): cookie=0x0, duration=13.535s, table=0,
n_packets=0, n_bytes=0, ip,nw_src=1.1.0.0/24 actions=output:1

```

The `show openflow instance test flows` command when executed on the switch displays the following:

Example

```

switch(vlan-3)# show openflow instance test

Configured OF Version      : 1.3 only
Negotiated OF Version      : 1.3
Instance Name              : test
Data-path Description      : test
Administrator Status       : Enabled
Member List                : VLAN 3
Pipeline Model              : Standard Match
Listen Port                : 6633
Operational Status         : Up
Operational Status Reason  : NA
Datapath ID                : 000340a8f09e8600

```



```

Mode : Active
Flow Location : Hardware and Software
No. of Hardware Flows : 0
No. of Software Flows : 0
Hardware Rate Limit : 0 kbps
Software Rate Limit : 100 pps
Conn. Interrupt Mode : Fail-Secure
Maximum Backoff Interval : 60 seconds
Probe Interval : 10 seconds
Hardware Table Miss Count : NA
No. of Software Flow Tables : 1
Egress Only Ports : None
Table Model : Policy Engine and Software
Source MAC Group Table : Disabled
Destination MAC Group Table : Disabled

```

```

Controller Id Connection Status Connection State Secure Role
-----
1 Connected Active Yes Equal

```

Virtualization mode versus Aggregation mode — VLAN tags in `packet_in` messages

There is a difference in the `packet_in` messages that are sent to the OpenFlow controller by the switch based on the mode that the OpenFlow instance is operating in. In Virtualization mode, no VLAN tags are sent in `packet_in` messages sent to the OpenFlow controller. Even if the packets that came into the switch on the OpenFlow instance had VLAN tags, the switch in `packet_in` messages sent to the controller removes the tags. Flows that match on VLAN PCP or modify VLAN PCP are not supported in Virtualization mode. Any tagged packets that are received in Virtualization mode may have their PCP modified to default. VLAN PCP is not matched because tag is always stripped in Virtualization mode.

In Aggregate mode, the switch always sends VLAN tags in `packet_in` messages sent to the OpenFlow controller. Even if the packets that came in to the switch on the OpenFlow instance did not have VLAN tags, the switch adds them in `packet_in` messages sent to the controller. The switch adds a VLAN tag either based on the tag that the packet already carried when it came in to the switch or based on the membership of the port that the packet came in to the switch.

Precedence level in meters

Standard-Match and IP-Control pipeline models

The DSCP Remark action in the meters associated with a flow for the above models writes the `prec_level` associated with the band directly into the DSCP field of the packet. This avoids increasing the drop precedence as defined in the OpenFlow specification.

Custom pipeline model

The DSCP Remark action in meters associated with a flow in custom pipeline model ignores the `prec_level` argument passed in the meter modification request. In this model, the remarking is performed based on the band to which the packet is classified into as per the defined rates.

Input DSCP	Green	Yellow	Red	Input DSCP	Green	Yellow	Red
0	0	0	0	31	17	9	1
1	1	1	1	32	32	24	16

Table Continued

2	2	2	2	33	33	25	17
3	3	3	3	34	34	26	18
4	4	4	4	35	35	27	19
5	5	5	5	36	36	28	20
6	0	0	0	37	37	29	21
7	1	1	1	38	24	16	8
8	8	0	0	39	25	17	9
9	9	1	1	40	40	32	24
10	10	2	2	41	41	33	25
11	11	3	3	42	42	34	26
12	12	4	4	43	43	35	27
13	13	5	5	44	44	36	28
14	0	0	0	45	45	37	29
15	1	1	1	46	32	24	16
16	16	8	0	47	33	25	17
17	17	9	1	48	48	40	32
18	18	10	2	49	49	41	33
19	19	11	3	50	50	42	34
20	20	12	4	51	51	43	35
21	21	13	5	52	52	44	36
22	8	0	0	53	53	45	37
23	9	1	1	54	40	32	24
24	24	16	8	55	41	33	25
25	25	17	9	56	56	48	40
26	26	18	10	57	57	49	41
27	27	19	11	58	58	50	42
28	28	20	12	59	59	51	43
29	29	21	13	60	60	52	44
30	16	8	0	61	61	53	45
				62	48	40	32
				63	49	41	33

Support for miss_len field in 'switch configuration' messages

The switch implementation does not honor the miss_len miss_send_len field specified in the packet-in switch configuration messages. This occurs because the switch does not buffer packets, and the controller sees the entire packet copied in packet-in message with buffer_id set as OFP_NO_BUFFER.

Once a controller deletes flows from Table 0, it has to re-add in order for traffic to flow through an OpenFlow switch

OpenFlow instance running Standard-Match Mode

- When the OpenFlow instance is configured with version 1.3, as soon as the controller connects and negotiates to 1.3, the switch internally adds the following Rule 1 on Table 0 (which says: GoTo Table 100):
 - Until the K/KA.15.15 release, if the controller were to delete Rule 1, there was no impact on packet forwarding on the switch for packets coming on OpenFlow VLANs. Also, the switch always rejected any attempts made by the controller to add back this default rule to Table 0.
 - Starting with the K/KA.15.16 release, if the controller were to delete Rule 1, there is an impact on packet forwarding on the switch for packets coming on OpenFlow VLANs. The switch drops packets incoming on OpenFlow VLANs. This dropping of packets continues until the controller adds the default rule back to Table 0. The switch complies to flow-mod requests from the controller on Table 0 based on its advertised capabilities.

OpenFlow instance running IP-Control Mode

- Rules on Table 0 in this mode:
 - Rule 1 -> All fields are wild-carded on Table 0 (which says Goto 100).
 - Rule 2 -> Match on Ether-Type IPv4 on Table 0 (which says Goto 50).
 - Rule 3 -> Match on Ether-Type IPv6 on Table 0 (which says Goto 50).
- Starting with the K/KA.15.16 release:
 - If Rule 1 is deleted, all non-IP traffic is dropped incoming on OpenFlow VLANs.
 - If Rule 2 is deleted, the switch drops all IPv4 traffic incoming on OpenFlow VLANs.
 - If Rule 3 is deleted, the switch drops all IPv6 traffic incoming on OpenFlow VLANs.
- There is also the case of a Table Miss Rule on Table 50:
Rule 4 -> All fields are wild-carded on Table 50 (which says Goto 102).
- Starting with the K/KA.15.16 release:
If Rule 4 is deleted, the switch drops all IP (both IPv4 and IPv6) traffic that does not match any of the rules on Table 50.
- In all these cases, the controller has to add these rules back to get the traffic to be forwarded as desired.

OpenFlow matching traffic destined for switch MAC address

When using OpenFlow, traffic that is destined to a routing switch that matches an OpenFlow flow that emulates routing does not get routed if there are no ARP entries on the switch for the devices involved. To trigger this, the traffic must be destined for a MAC address assigned on the switch.

In traditional networking without OpenFlow, when packets to a routable IP destination arrive, the packets will be buffered/dropped until the next hop to that IP destination is resolved. The ARP resolution is triggered by the switch itself.



NOTE: To make this work with OpenFlow, you must ensure that ARPs to all hosts are resolved.

OpenFlow custom pipeline implementation notes

- OpenFlow Table modifications cannot be done in v1/v2 modules or on v3 modules in compatibility mode.
- When the instance is configured in Standard match mode or ip-control-table-mode, controllers cannot change the default pipeline advertised by controller.
- For IPv6 packets, MAC SA modification is not allowed. L4 ports modifications are not allowed for IPv6 packets.
- Write-metadata instructions are not supported.
- Rules with actions as packet-modifications and OFPP_CONTROLLER are directly rejected. However, if the rules span across multiple tables (write-instruction in first table having packet modifications and write-instructions in further tables having OFPP_CONTROLLER) would be accepted – however, the packets taken to CPU/CONTROLLER are unmodified.
- Special ports OFPP_ALL, OFPP_LOCAL, OFPP_TABLE, and OFPP_IN_PORT are not supported.
- Write-instruction having OFPP_CONTROLLER cannot have a Goto instruction associated with it.
- Write-instruction rules must always have output port specified or Goto to next table or “output port + goto”.
- Counters are associated with tables and flow entries if available. Otherwise, the table or flow creation is rejected with an error `Resources not available` to the controller.
- Tables configured from the controller must have a minimum of 512 entries or the table creation request is rejected.
- Before a table is deleted, the controller must remove all flows in previous tables pointing to the deleted table. If all flows are not removed, the controller table modification (deletion/modification) is not allowed.
- For multiple custom-pipeline instances, the user must reduce the size of the tables in the first custom-pipeline instance. Following this procedure, the user must disable and enable the other custom-match instances for them to become operational.



NOTE:

When a same OpenFlow meter is used two or more times in an OpenFlow pipeline, it results in skewed meter rates leading to unpredictable behavior in how the packets are metered. HPE recommends that you do not use a meter more than once in a packet pipeline to avoid undesired behaviors.

Multi-VLAN implementation notes

- For a Multi-VLAN instance, the first VLAN in the Multi-VLAN instance is part of the DPID. Since VLANs cannot be part of multiple instances, which ensures a unique DPID for each instance.
- A VLAN, which is member VLAN of one instance, cannot be member VLAN of another instance.
- A VLAN cannot be dynamically added to an instance when it is enabled, the instance must be disabled before adding a VLAN to an OpenFlow instance.
- The user can configure VLANs that are member VLANs of an instance after creating and enabling the Multi-VLAN instance.
- At least one Member-VLAN of a Multi-VLAN instance must be configured on the switch else the `operStatus` is marked `DOWN`.
- Management and controller VLANs cannot be part of an instance.

Implementation notes for OpenFlow groups in hardware

- Groups are not supported in hardware when switch is running in v1 mode.
- Special ports `OFPP_ALL`, `OFPP_NORMAL`, `OFPP_TABLE`, `OFPP_IN_PORT`, `OFPP_LOCAL` and `OFPP_CONTROLLER` are not supported in OpenFlow groups in hardware.
- Tunnels cannot be part of OpenFlow ALL group.
- Rules in OpenFlow software tables can refer to Groups in software or hardware.
- Rules in OpenFlow hardware tables cannot refer to groups in software.
- On custom pipeline instances, groups in hardware have counters subject to the availability of the counters.
- If the group is in software, it can be referenced only from a software table.
- Rules in hardware tables with group action must not have a goto instruction associated with it.
- Group Modifications:
 - Modification of a group in hardware that leads to having no Ports, is not allowed.
 - If the resultant modification can no longer be accommodated in H/W, group modification requests for Groups in H/W are rejected.
- In custom mode, the groups are always in hardware. Attempts to add groups with actions other than output ports are rejected.
- The number of OpenFlow groups that can be created on an OpenFlow instance is 1024.

Implementation notes

- **Limitations with `APPLY {OUTPUT=CTRL}` + `APPLY {OUTPUT=NORMAL}`**

When the switch executes the `output=ctrl` as an apply action, any `output=normal` action that follows does not work if the packet is destined to the switch.

As a work around to this limitation, the controller must use **WRITE {OUTPUT=CTRL}** to copy the packets to the controller.

- When the system time is updated, the creation time of all the OpenFlow entities such as Flows, Groups, Meters, and Ports are reset to the new time. Any time based operations such as calculation of duration, idle and hard timeout are restarted based on the new time.
- Packets with source MAC address as a multicast MAC address are not forwarded from Table 100 to Table 200, even if the specified rule has a GOTO action.
- Rules which match on overridden protocols in the OpenFlow exclusion list do not support **APPLY/WRITE {OUTPUT=NORMAL}** action following er any other OUTPUT action.
- Rules with **APPLY/WRITE {OUTPUT=(FLOOD|NORMAL)}** are not supported after a **{OUTPUT=IN_PORT}** action.
- Rules with action combination **APPLY/WRITE {SET-FIELD (ETH_DST + IP_SRC||IP_DST)} + {OUTPUT =NORMAL}** do not modify the **ETH_DST** to the value specified in the packet.
- OpenFlow instances running in custom pipeline model do not perform any OpenFlow lookups for packets coming from the switch CPU.
- OpenFlow forces a fresh switch forwarding plane lookup for packets executing the action sequence **APPLY/WRITE {SET-FIELD (IP_SRC||IP_DST) + [SET-FIELD ...]} + {OUTPUT=NORMAL}**

Switches running OpenFlow can securely connect to HPE VAN SDN controller.

To accomplish the secure connection, follow these procedures:

Procedure

1. On the Switch running OpenFlow, create a TA (Trusted Anchor) profile: `crypto pki ta-profile VanProfile [TA-PROFILE-NAME]`

2. Copy root certificate to the switch:

```
copy tftp ta-certificate [TA-PROFILE-NAME] [IP-ADDRESS of the server] [FILE-NAME]
```

3. Create an identity profile on the switch:

```
crypto pki identity-profile [PROFILE-NAME-STR] subject common-name [CN-VALUE]
```

4. Make a certificate signing request:

```
crypto pki create-csr certificate-name [CERT-NAME] ta-profile [TA-PROFILE-NAME] usage [openflow]
```

The same root certificate installed on the switch in step 2 must sign the CSR generated in this step.

5. Install the leaf certificate:

```
crypto pki install-signed-certificate
```

6. Paste the contents of the signed certificate in PEM format into the switch console.



NOTE: Apart from Steps 5 and 6, another way to install PEM formatted certificate is to download it via TFTP using the command:

```
copy tftp local-certificate [TFTP Server IPv4/IPv6 address] [Name of the file containing certificate in PEM format]
```

7. Configure OpenFlow to connect to the VAN SDN controller:

```
openflow
controller-id 3 ip 103.0.11.31 port 6634 controller interface
vlan 1 instance "van"
member vlan 100
controller id 3 secure
version 1.3
limit hardware-rate 10000000
limit software-rate 10000
enable
```

```
exit  
enable
```


For information on tunnels and service insertion, see the *Service Insertion Guide*.