Hewlett Packard
Enterprise

HPE FlexNetwork MSR Router Series

Comware 7 OpenFlow Configuration Guide

Acknowledgments

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft® and Windows® are trademarks of the Microsoft group of companies.

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.

Contents

Configuring OpenFlow

OpenFlow is the communications interface defined between the control and forwarding layers of a Software-Defined Networking architecture. With OpenFlow, you can perform centralized data forwarding management for physical and virtual devices through controllers.

The term "switch" in this chapter refers to routers that support OpenFlow.

Overview

OpenFlow separates the data forwarding and routing decision functions. It keeps the flow-based forwarding function and employs a separate controller to make routing decisions. An OpenFlow switch communicates with the controller through an OpenFlow channel. An OpenFlow channel can be encrypted by using TLS or run directly over TCP. An OpenFlow switch exchanges control messages with the controller through an OpenFlow channel to perform the following tasks:

- Receive flow table entries or data from the controller.
- Report information to the controller.

Unless otherwise stated, a switch refers to an OpenFlow switch throughout this document.

Figure 1 OpenFlow network diagram
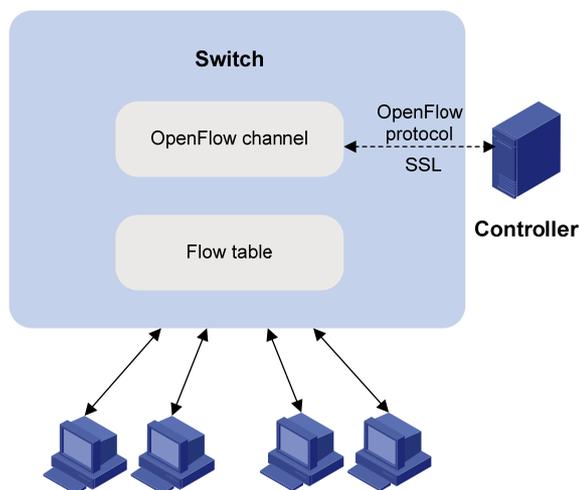
OpenFlow switch

OpenFlow switches include the following types:

- **OpenFlow-only**—Supports only OpenFlow operation.
- **OpenFlow-hybrid**—Supports both OpenFlow operation and traditional Ethernet switching operation.

OpenFlow port

OpenFlow supports the following types of ports:

- **Physical port**—Corresponds to a hardware interface, such as an Ethernet interface. A physical port can be either an ingress port or an output port.

- **Logical port**—Does not correspond to a hardware interface and might be defined by non-OpenFlow methods. For example, aggregate interfaces and tunnel interfaces are logical ports. A logical port can be either an ingress port or an output port.
- **Reserved port**—Defined by OpenFlow to specify forwarding actions. Reserved ports include the following types:
  - **All**—All ports that can be used to forward a packet.
  - **Controller**—OpenFlow controller.
  - **Table**—Flow table.
  - **In port**—Packet ingress port.
  - **Any**—Generic port description.
  - **Local**—Local CPU.
  - **Normal**—Normal forwarding process.
  - **Flood**—Flooding.

  Except the **Any** type, all reserved ports can be used as output ports. Only the **Controller** and **Local** types can be used as ingress ports.

OpenFlow instance

Unless otherwise stated, an OpenFlow switch refers to an OpenFlow instance throughout this document.

You can configure one or more OpenFlow instances on the same device. A controller considers each OpenFlow instance as a separate OpenFlow switch and deploys forwarding instructions to it.

OpenFlow instance mode

When the global mode is enabled for an OpenFlow instance, the flow entries take effect on packets within the network.

When the port mode is enabled for an OpenFlow instance, the flow entries take effect on packets only on ports that are bound to the OpenFlow instance.

Activation and reactivation

The configurations for an OpenFlow instance take effect only after the OpenFlow instance is activated.

The controller can deploy flow entries to an OpenFlow instance only after the OpenFlow instance reports the following device information to the controller:

- Support capabilities for OpenFlow.
- Information about ports that belongs to the OpenFlow instance.

An activated OpenFlow instance must be reactivated when any of the OpenFlow instance configurations are changed.

After reactivation, the OpenFlow instance is disconnected from all controllers and then reconnected to them.

OpenFlow instance port

An OpenFlow switch sends information about the following ports to the controller:

- Physical ports.
- Logical ports.
- Reserved ports of the **Local** type.

OpenFlow flow table

An OpenFlow switch matches packets against one or more flow tables. A flow table contains flow entries, and packets are matched based on the matching precedence of flow entries.

OpenFlow flow tables include the following types:

- **MAC-IP**—Combines the MAC address table and FIB table.

  A MAC-IP flow table provides the following match fields:
  - Destination MAC address.
  - VLAN.
  - Destination IP address.

  A MAC-IP flow table provides the following actions:
  - Modifying the destination MAC address.
  - Modifying the source MAC address.
  - Modifying the VLAN.
  - Modifying the tunnel ID.
  - Specifying the output port.

  For more information, see "Appendix B MAC-IP flow table."

- **Extensibility**—Combines Ternary Content Addressable Memory (TCAM) tables and tables programmed by software.

Flow entry

Figure 2 Flow entry components

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

A flow entry contains the following fields:

- **Match fields**—Matching rules of the flow entry. These contain the ingress port, packet headers, and metadata specified by the previous table.

- **Priority**—Matching precedence of the flow entry. When a packet is matched against the flow table, only the highest priority flow entry that matches the packet is selected.

- **Counters**—Counts of the packets that match the flow entry.

- **Instructions**—Used to modify the action set or pipeline processing. Instructions include the following types:
  - **Meter**—Directs the packets to the specified meter to rate limit the packets.
  - **Apply-Actions**—Applies the specified actions in the action list immediately.
  - **Clear-Actions**—Clears all actions in the action set immediately.
  - **Write-Actions**—Modifies all actions in the action set immediately.
  - **Write-Metadata**—Modifies packets between two flow tables if there are multiple flow tables.
  - **Goto-Table**—Indicates the next flow table in the processing pipeline.

  Actions are executed in one of the following ways:
  - **Action Set**—When the instruction set of a flow entry does not contain a **Goto-Table** instruction, pipeline processing stops. Then, the actions in the action set are executed in the order specified by the instruction list. An action set contains a maximum of one action of each type.
  - **Action List**—The actions in the action list are executed immediately in the order specified by the action list. The effect of those actions is cumulative.

  Actions include the following types:

- - **(Required.) Output**—The Output action forwards a packet to the specified OpenFlow port. OpenFlow switches must support forwarding packets to physical ports, logical ports, and reserved ports.
  - **(Required.) Drop**—There is no explicit action to represent drops. Packets whose action sets have no output actions are dropped. Typically, packets are dropped due to empty instruction sets, empty action sets, or the executing a Clear-Actions instruction.
  - **(Required.) Group**—Process the packet through the specified group. The exact interpretation depends on group type.
  - **(Optional.) Set-Queue**—The Set-Queue action sets the queue ID for a packet. When the packet is forwarded to a port by using the output action, the queue ID determines which queue attached to this port is used for scheduling and forwarding the packet. The forwarding behavior is dictated by the configuration of the queue and is used to provide basic QoS support.
  - **(Optional.) Push-Tag/Pop-Tag**—Switches support the ability to push or pop tags, such as VLAN tags, MPLS tags, and PBB tags.
  - **(Optional.) Set-Field**—The Set-Field actions are identified by their field type and modify the values of corresponding header fields in the packet. Set-Field actions are always applied to the outermost header. For example, a Set VLAN ID action always sets the ID of the outermost VLAN tag.
  - **(Optional.) Change-TTL**—The Change-TTL actions modify the values of the IPv4 TTL, IPv6 Hop Limit, or MPLS TTL in the packet. Change-TTL actions are always applied to the outermost header. The Change-TTL actions include the following the actions: Set TTL, Decrement TTL, and Copy TTL (outwards or inwards).
- **Timeouts**—Maximum amount of idle time or hard time for the flow entry.
  - **idle time**—The flow entry is removed when it has matched no packets during the idle time.
  - **hard time**—The flow entry is removed when the hard time timeout is exceeded, regardless of whether or not it has matched packets.
- **Cookie**—Flow entry identifier specified by the controller.

OpenFlow pipeline

The OpenFlow pipeline processing defines how packets interact with flow tables contained by a switch.

The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. The packet is first matched against flow entries of the first flow table, which is flow table 0. A flow entry can only direct a packet to a flow table number that is greater than its own flow table number.

When a packet matches a flow entry, the OpenFlow switch updates the action set for the packet and passes the packet to the next flow table. In the last flow table, the OpenFlow switch executes all actions to modify packet contents and specify the output port for packet forwarding. If the instruction set of a flow table contains an action list, the OpenFlow switch immediately executes the actions for a copy of the packet in this table.

Figure 3 OpenFlow forwarding workflow

4
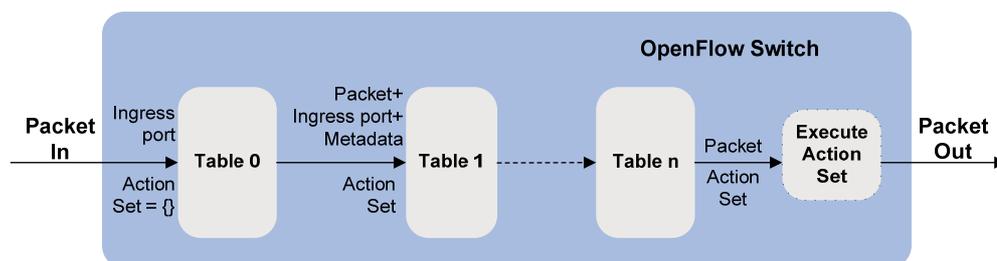
Table-miss flow entry

Every flow table must support a table-miss flow entry to process table misses. The table-miss flow entry specifies how to process packets that were not matched by other flow entries in the flow table.

The table-miss flow entry wildcards all match fields (all fields omitted) and has the lowest priority 0.

The table-miss flow entry behaves in most ways like any other flow entry.

Group table

The ability for a flow entry to point to a group enables OpenFlow to represent additional methods of forwarding. A group table contains group entries.

Figure 4 Group entry components

Group Identifier	Group Type	Counters	Action Buckets

A group entry contains the following fields:

- **Group Identifier**—A 32 bit unsigned integer uniquely identifying the group.
- **Group Type**—Type of the group:
  - **All**—Execute all buckets in the group. This group is used for multicast or broadcast forwarding.
  - **Select**—Execute one bucket in the group.
  - **Indirect**—Execute the one defined bucket in the group.
  - **Fast failover**—Execute the first live bucket.
- **Counters**—Updated when packets are processed by a group.
- **Action Buckets**—An ordered list of action buckets, where each action bucket contains a set of actions to execute and associated parameters.

Meter table

Meters enable OpenFlow to implement various simple QoS operations, such as rate-limiting. A meter table contains meter entries.

Figure 5 Meter entry components

Meter Identifier	Meter Bands	Counters

A meter entry contains the following fields:

- **Meter Identifier**—A 32 bit unsigned integer uniquely identifying the meter.
- **Meter Bands**—Each meter can have one or more meter bands. Each band specifies the rate at which the band applies and the way packets should be processed. If the current rate of packets exceeds the rate of multiple bands, the band with the highest configured rate is used.
- **Counters**—Updated when packets are processed by a meter.

Figure 6 Band components

Band Type	Rate	Counters	Type Specific arguments

A meter band contains the following fields:

- **Band Type**—(Optional.) Packet processing methods. Options are:

- o **Drop**—Discards the packet when the rate of the packet exceeds the band rate.
- o **DSCP Remark**—Remarks the DSCP field in the IP header of the packet.
- **Rate**—Defines the lowest rate at which the band can apply.
- **Counters**—Updated when packets are processed by a band.
- **Type Specific Arguments**—Some band types have specific arguments.

OpenFlow channel

The OpenFlow channel is the interface that connects each OpenFlow switch to a controller. The controller uses the OpenFlow channel to exchange control messages to configure and manage the switch, receive events from the switch, and send packets out the switch. The OpenFlow channel is usually encrypted by using TLS. Also, an OpenFlow channel can be run directly over TCP.

The OpenFlow protocol supports the following message types: controller-to-switch, asynchronous, and symmetric. Each message type has its own subtypes.

Controller-to-Switch messages

Controller-to-Switch messages are initiated by the controller and used to directly manage or inspect the state of the switch. Controller-to-Switch messages might or might not require a response from the switch.

The Controller-to-Switch messages include the following subtypes:

- **Features**—The controller requests the basic capabilities of a switch by sending a features request. The switch must respond with a features reply that specifies the basic capabilities of the switch.
- **Configuration**—The controller sets and queries configuration parameters in the switch. The switch only responds to a query from the controller.
- **Modify-State**—The controller sends Modify-State messages to manage state on the switches. Their primary purpose is to add, delete, and modify flow or group entries in the OpenFlow tables and to set switch port properties.
- **Read-State**—The controller sends Read-State messages to collect various information from the switch, such as current configuration and statistics.
- **Packet-out**—These are used by the controller to send packets out of the specified port on the switch, or to forward packets received through Packet-in messages. Packet-out messages must contain a full packet or a buffer ID referencing a packet stored in the switch. The message must also contain a list of actions to be applied in the order they are specified. An empty action list drops the packet.
- **Barrier**—Barrier messages are used to confirm the completion of the previous operations. The controller send s Barrier request. The switch must send a Barrier reply when all the previous operations are complete.
- **Role-Request**—Role-Request messages are used by the controller to set the role of its OpenFlow channel, or query that role. It is typically used when the switch connects to multiple controllers.
- **Asynchronous-Configuration**—These are used by the controller to set an additional filter on the asynchronous messages that it wants to receive, or to query that filter. It is typically used when the switch connects to multiple controllers.

Asynchronous messages

Switches send asynchronous messages to controllers to inform a packet arrival or switch state change. For example, when a flow entry is removed due to timeout, the switch sends a flow removed message to inform the controller.

The asynchronous messages include the following subtypes:

- **Packet-In**—Transfer the control of a packet to the controller. For all packets forwarded to the Controller reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers. Other processing, such as TTL checking, can also generate packet-in events to send packets to the controller. The Packet-in events can include the full packet or can be configured to buffer packets in the switch. If the packet-in event is configured to buffer packets, the packet-in events contain only some fraction of the packet header and a buffer ID. The controller processes the full packet or the combination of the packet header and the buffer ID. Then, the controller sends a Packet-out message to direct the switch to process the packet.
- **Flow-Removed**—Inform the controller about the removal of a flow entry from a flow table. These are generated due to a controller flow delete request or the switch flow expiry process when one of the flow timeouts is exceeded.
- **Port-status**—Inform the controller of a state or setting change on a port.
- **Error**—Inform the controller of a problem or error.

Symmetric messages

Symmetric messages are sent without solicitation, in either direction.

The symmetric messages contain the following subtypes:

- **Hello**—Hello messages are exchanged between the switch and controller upon connection startup.
- **Echo**—Echo request or reply messages can be sent from either the switch or the controller, and must return an echo reply. They are mainly used to verify the liveness of a controller-switch connection, and might also be used to measure its latency or bandwidth.
- **Experimenter**—This is a staging area for features meant for future OpenFlow revisions.

Protocols and standards

OpenFlow Switch Specification Version 1.3.3

OpenFlow configuration task list

Tasks at a glance

(Required.) Configure an OpenFlow instance:
1. (Required.) Creating an OpenFlow instance
2. (Required.) Configuring an OpenFlow instance:
  o (Required.) Configuring the OpenFlow instance mode
  o (Optional.) Creating flow tables for an OpenFlow instance
  o (Optional.) Setting the controller mode
  o (Optional.) Setting the maximum number of flow entries for an extensibility flow table
  o (Optional.) Configuring the datapath ID
  o (Optional.) Enabling an SSL server for an OpenFlow instance
  o (Optional.) Configuring the default action of table-miss flow entries
  o (Optional.) Forbidding an OpenFlow instance to report the specified types of ports to controllers
  o (Optional.) Binding an OpenFlow instance to ports
  o (Optional.) Binding an port to an OpenFlow instance
3. (Required.) Activating or reactivating an OpenFlow instance

(Required.) Configure controllers for an OpenFlow switch: • (Required.) Configuring controllers for an OpenFlow switch • (Optional.) Configuring controllers and auxiliary connections • (Optional.) Setting the connection interruption mode

Tasks at a glance
(Optional.) Setting OpenFlow timers
(Optional.) Configuring OpenFlow to support dynamic MAC addresses

Configuring OpenFlow instances

Creating an OpenFlow instance

> **(!) IMPORTANT:**
>
> The OpenFlow OAP instance does not support the output action of sending packets to the controller.

To configure an OpenFlow instance or the OpenFlow OAP instance:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an OpenFlow instance or the OpenFlow OAP instance, and enter OpenFlow instance or OpenFlow OAP instance view.	**openflow instance** { *instance-id* | **oap instance**}	By default, no OpenFlow instance or OpenFlow OAP instance exists.
3. (Optional.) Specify a description for the OpenFlow instance.	**description** *text*	By default, an OpenFlow instance does not have a description.

Configuring the OpenFlow instance mode

> **(!) IMPORTANT:**
>
> The OpenFlow OAP instance does not support this feature.

To configure the OpenFlow instance mode:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Configure the OpenFlow instance mode.	**classification** { **global** | **port** }	By default, the OpenFlow instance mode is not configured.

Creating flow tables for an OpenFlow instance

> **(!) IMPORTANT:**
>
> The OpenFlow OAP instance does not support this feature.

You can create one MAC-IP flow table and one or more extensibility flow tables for an OpenFlow instance.

To create flow tables for an OpenFlow instance:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Create flow tables for the OpenFlow instance.	**flow-table** { **extensibility** *table-id* | **mac-ip** *table-id* }&<1-n>	By default, an OpenFlow instance contains one extensibility flow table with an ID of 0.

Setting the controller mode

> ⚠ **IMPORTANT:**
> The OpenFlow OAP instance does not support this feature.

An OpenFlow instance can connect to one or more controllers, depending on the controller mode the OpenFlow instance uses:

- **Single**—The OpenFlow instance connects to only one controller at a time. When communication with the current controller fails, the OpenFlow instance uses another controller.
- **Multiple**—The OpenFlow instance can simultaneously connect to multiple controllers. When communication with any controller fails, the OpenFlow instance attempts to reconnect to the controller after a reconnection interval.

To set the controller mode for an OpenFlow instance:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Set the controller mode.	**controller mode** { **multiple** | **single** }	By default, the **multiple** mode is used.

Setting the maximum number of flow entries for an extensibility flow table

You can set the maximum number of flow entries that each extensibility flow table supports. When the maximum number is reached, the OpenFlow instance does not accept new flow entries for that table and sends a deployment failure notification to the controller.

To set the maximum number of flow entries for an extensibility flow table:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance or OpenFlow OAP instance view.	**openflow instance** { *instance-id* | **oap-instance** }	N/A
3. Set the maximum number of flow entries for an extensibility flow table.	**flow-entry max-limit** *limit-value*	By default, an extensibility flow table can have a maximum of 65535 flow entries.

Configuring the datapath ID

The datapath ID is a unique identifier for an OpenFlow switch (OpenFlow instance). Do not configure the same datapath ID for different OpenFlow switches.

To configure the datapath ID:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view or OpenFlow OAP instance view.	**openflow instance** *instance-id* { *instance-id* | **oap-instance** }	N/A
3. Configure the datapath ID.	**datapath-id** *id*	By default, the datapath ID of an OpenFlow instance contains the instance ID and the bridge MAC address of the device. The upper 16 bits are the instance ID and the lower 48 bits are the bridge MAC address of the device.

Enabling an SSL server for an OpenFlow instance

Typically, an OpenFlow instance actively connects to the controller acting as a TCP/SSL client. An OpenFlow instance acts as an SSL client when an SSL server is enabled for the controller.

You can configure this feature to enable an SSL server for an OpenFlow instance. After the SSL server is enabled for an OpenFlow instance, the controller acts as the SSL client and actively connects to the OpenFlow instance.

You can enable only one SSL server for an OpenFlow instance.

To enable an SSL server for an OpenFlow instance:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view or OpenFlow OAP instance view.	**openflow instance** *instance-id* { *instance-id* | **oap-instance** }	N/A
3. Enable an SSL server for an OpenFlow instance.	**listening port** *port-number* **ssl** *ssl-policy-name*	By default, the SSL server is disabled for an OpenFlow instance.

Configuring the default action of table-miss flow entries

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view or OpenFlow OAP instance view.	**openflow instance** *instance-id* { *instance-id* | **oap-instance** }	N/A
3. Configure the default action of table-miss flow entries to forward packets to the normal pipeline.	**default table-miss permit**	By default, the default action of table-miss flow entries is to drop packets.

Forbidding an OpenFlow instance to report the specified types of ports to controllers

You can configure this feature to forbid an OpenFlow instance to report information of the specified VLAN or VSI interfaces to controllers.

To forbid an OpenFlow instance to report the specified types of ports to controllers:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view	**openflow instance** *instance-id*	N/A
3. Forbid an OpenFlow instance to report the specified types of ports to controllers.	**forbidden port** { **vlan-interface** | **vsi-interface** } *	By default, all ports that belong to an OpenFlow instance are reported to the controllers.

Binding an OpenFlow instance to ports

Before you configure this feature for an OpenFlow instance, first enable the port mode for the OpenFlow instance.

To bind an OpenFlow instance to ports:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Bind the OpenFlow instance to ports.	**forbidden port** { **vlan-interface** | **vsi-interface** } *	By default, an OpenFlow instance is not bound to ports.

Binding an port to an OpenFlow instance

Before you configure this feature for an OpenFlow instance, first enable the port mode for the OpenFlow instance.

To bind a port to an OpenFlow instance:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Bind the port to an OpenFlow instance.	**openflow instance** *instance-id*	By default, a port is not bound to an OpenFlow instance.

Activating or reactivating an OpenFlow instance

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view or OpenFlow OAP instance view.	**openflow instance** *instance-id* { *instance-id*	**oap-instance** }	N/A
3. Activate or reactivate the OpenFlow instance.	**active instance**	By default, an OpenFlow instance is not activated.

Configuring controllers for an OpenFlow switch

> ⊘ **IMPORTANT:**
> The OpenFlow OAP instance does not support this feature.

A switch can establish connections with multiple controllers. The controller role contains the following types:

- **Equal**—In this role, the controller has full access to the switch and is equal to other controllers in the same role. By default, the controller receives all switch asynchronous messages such as packet-in and flow removed messages. The controller can send controller-to-switch messages to modify the state of the switch.

- **Master**—This role is similar to the Equal role and has full access to the switch. The difference is that up to one controller in this role is allowed for a switch.

- **Slave**—In this role, the controller has read-only access to the switch.

  The controller cannot send controller-to-switch messages to perform the following tasks:

  - Deploy flow entries, group entries, and meter entries.
  - Modify the port and switch configurations.
  - Send Packet-out messages.

  By default, the controller does not receive switch asynchronous messages except Port-status messages. The controller can send Asynchronous-Configuration messages to set the asynchronous message types it wants to receive.

When OpenFlow operation is initiated, a switch is simultaneously connected to multiple controllers in Equal state. A controller can request its role to be changed at any time.

Configuring controllers and main connections

A switch can establish connections with multiple controllers. The OpenFlow channel between the OpenFlow switch and each controller can have only one main connection. The main connection processes control messages to complete tasks such as deploying entries, obtaining data, and sending information. The main connection must be a reliable connection using TCP or SSL.

To specify a controller for an OpenFlow switch and configure the main connection to the controller:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Specify a controller and	**controller** *controller-id* **address**	By default, an OpenFlow instance

Step	Command	Remarks
configure the main connection to the controller.	{ **ip** *ip-address* | **ipv6** *ipv6-address* } [**port** *port-number*] [**local address** { **ip** *local-ip-address* | **ipv6** *local-ipv6-address* } [**port** *local-port-number*]] [**ssl** *ssl-policy-name*] [**vrf** *vrf-name*]	is not configured with any main connection. As a best practice, configure a unicast IP address for a controller. Otherwise, an OpenFlow switch might fail to establish a connection with the controller. As a best practice, configure a unicast source IP address that is the IP address of a port belonging to an OpenFlow instance. Otherwise, the OpenFlow switch might fail to establish a connection with the controller.

Configuring controllers and auxiliary connections

The OpenFlow channel might have one main connection and multiple auxiliary connections. Auxiliary connections are used to improve the communication performance between the controller and OpenFlow switches.

An auxiliary connection can have the different destination IP address and port number from the main connection. If no destination IP address and port number are specified, the auxiliary connection uses the destination IP address and port number configured for the main connection.

Make sure the configuration of an auxiliary connection does not conflict with that of the main connection. Otherwise, the auxiliary connection cannot be established.

To specify a controller for an OpenFlow switch and configure an auxiliary connection to the controller:

	Step	Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3.	Specify a controller and configure an auxiliary connection to the controller.	**controller** *id* **auxiliary** *auxiliary-id* **transport** { **tcp** | **udp** | **ssl** *ssl-policy-name* } [**address** { **ip** *ip-address* | **ipv6** *ipv6-address* }] [**port** *port-number*]	By default, an OpenFlow instance is not configured with any auxiliary connection.

Setting the connection interruption mode

When an OpenFlow switch is disconnected from all controllers, the OpenFlow switch is set to either of the following modes:

- **Secure**—The OpenFlow switch forwards traffic based on flow tables and does not remove unexpired flow entries.
- **Standalone**—The OpenFlow switch uses the normal forwarding process.

The OpenFlow switch forwards traffic based on flow tables when it reconnects to a controller successfully.

To set the connection interruption mode for an OpenFlow switch:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Set the connection interruption mode.	**fail-open mode** { **secure** | **standalone** }	By default, the **secure** mode is used when an OpenFlow instance is established, and the controller deploys the table-miss flow entry (the action is **Drop**) to the OpenFlow instance.

Setting OpenFlow timers

> **IMPORTANT:**
> The OpenFlow OAP instance does not support the reconnection interval.

An OpenFlow switch supports the following timers:

- **Connection detection interval**—Interval at which the OpenFlow switch sends an Echo Request message to a controller. When the OpenFlow switch receives no Echo Reply message within three intervals, the OpenFlow switch is disconnected from the controller.
- **Reconnection interval**—Interval for the OpenFlow switch to wait before it attempts to reconnect to a controller.

To set OpenFlow timers for an OpenFlow switch:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view or OpenFlow OAP instance view.	**openflow instance** *instance-id* { *instance-id* | **oap-instance** }	N/A
3. Set the connection detection interval.	**controller echo-request interval** *time-value*	The default setting is 5 seconds.
4. Set the reconnection interval.	**controller connect interval** *time-value*	The default setting is 60 seconds.

Configuring OpenFlow to support dynamic MAC addresses

> **IMPORTANT:**
> The OpenFlow OAP instance does not support this feature.

On an OpenFlow switch that supports MAC-IP flow tables, you can configure OpenFlow to support querying and deleting dynamic MAC addresses in the flow tables.

To configure OpenFlow to support dynamic MAC addresses:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id* { *instance-id* | **oap-instance** }	N/A
3. Configure OpenFlow to support dynamic MAC addresses.	**mac-ip dynamic-mac aware**	By default, OpenFlow prohibits controllers from querying and deleting dynamic MAC addresses.

Displaying and maintaining OpenFlow

Execute **display** commands in any view.

Task	Command
Display the detailed information for an OpenFlow instance.	**display openflow instance** [*instance-id* | **oap-instance**]
Display flow table entries for an OpenFlow instance.	**display openflow instance** { *instance-id* | **oap-instance** } **flow-table** [*table-id*]
Display controller information for an OpenFlow instance.	**display openflow instance** { *instance-id* { **controller** [*controller-id*] | **listened** } | **oap-instance listened** }
Display group information for an OpenFlow instance.	**display openflow instance** { *instance-id* | **oap-instance** } **group** [*group-id*]
Display meter information for an OpenFlow instance.	**display openflow instance** { *instance-id* | **oap-instance** } **meter** [*meter-id*]
Display summary OpenFlow instance information.	**display openflow summary**
Display OAP context information	**display openflow oap-context** [**oap-interface** *oap-interface-type oap-interface-number* | **in-interface** *in-interface-type in-interface-number* | **out-interface** *out-interface-type out-interface-number*]
Display auxiliary connection information and statistics about received and sent packets.	**display openflow instance** *instance-id* **auxiliary** [*controller-id* [**auxiliary** *auxiliary-id*]]
Clear statistics on packets that a controller sends and receives for an OpenFlow instance.	**reset openflow instance** { *instance-id* { **controller** [*controller-id*] | **listened** } | **oap-instance listened** } **statistics**

OpenFlow configuration example

Network requirements

As shown in , perform the following tasks on the router:
- Create OpenFlow instance 1, and activate the OpenFlow instance.
- Configure the IP address for controller 1 to have the controller manage the router.

Figure 7 Network diagram
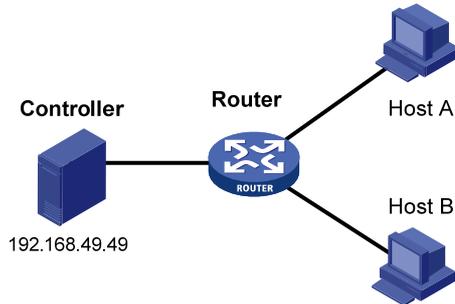
Configuration procedure

Create OpenFlow instance 1 and enable the global mode for the OpenFlow instance.

```
[Router] openflow instance 1
[Router-of-inst-1] classification global
```

Specify controller 1 for OpenFlow instance 1 and activate the instance.

```
[Router-of-inst-1] controller 1 address ip 192.168.49.49
[Router-of-inst-1] active instance
```

Verifying the configuration

Display detailed information about the OpenFlow instance.

```
[Router-of-inst-1] display openflow instance 1
Instance 1 verbose information:

Configuration information:
 Description   : --
 Active status : Active
 Inactive configuration:
  None
 Active configuration:
  Classification: Global(Standard)
   4092, 4094
  In-band management VLAN, total VLANs(0)
   empty VLAN
  Connect mode: multiple
  Mac-address learning: permit
  Flow table:
   Table ID(type): 0(Extensibility), count: 0
  Flow-entry max-limit: 65535
  Datapath ID: 0x0064001122000101
Port information:
 GigabitEthernet1/0/3
Active channel information:
 Controller 1 IP address: 192.168.49.49 port: 6633
```

Appendixes

Appendix A Application restrictions

Matching restrictions

Protocol packet matching

If protocols are enabled, protocol packets (except LLDP frames) are processed by the corresponding protocols instead of the OpenFlow protocol.

For more information about LLDP frame matching, see "LLDP frame matching."

Metadata matching

Metadata is used to pass matching information between flow tables. The controller deploys metadata matching entries only to non-first flow tables. If the controller deploys a metadata matching entry to the first flow table, the switch returns an unsupported flow error.

Instruction restrictions

Table 1 Instruction restrictions

Instruction type	Restrictions
Clear-Actions	The Clear-Actions instruction has the following restrictions: • For the single flow table, the flow entries of the table cannot include this instruction and other instructions at the same time. • For multiple flow tables of the pipeline, only the flow entries of the first flow table can include this instruction and other instructions at the same time.
Apply-Actions	The action list of the Apply-Actions instruction cannot include multiple Output actions. When the action list of the Apply-Actions instruction includes only one Output action, the switch processes the action list as described in "Restrictions for merging the action list into the action set."
Write-Metadata/mask	The flow entries of the last table of the pipeline cannot include this instruction. Otherwise, the switch returns an unsupported flow error.
Goto-Table	

Restrictions for merging the action list into the action set

The switch follows the following restrictions to merge the action list into the action set:

- When the action set and the action list do not contain the Output or Group action:
  - If the actions in the action set do not conflict with the actions in the action list, the switch merges the action list into the action set.
  - If the actions in the action set conflicts with the actions in the action list, actions in the action list are replaced with the actions in the action set.
- When the action set and the action list contain the Output action or the Group action:
  - If both the action list and the action set contain an Output action, the Output action in the action list takes precedence over the Output action in the action set. The Output action in

17

the action list does not modify the packet. The Output action in the action set is executed at the last step of the pipeline processing to modify the packet.

- o If either the action list or the action set contains an Output action, the port specified by the Output action is treated as the output port of the packet. The actions are executed in the order defined by the action set rules.
- o If the action list contains an Output action the action set contains a Group action, the Output action does not modify the packet and the Group action is executed.

Packet-out messages restrictions

Ingress port

If a Normal, Local, In Port, or Controller reserved port is used as the output port in a Packet-out message, the ingress port must be a physical port or logical port.

Buffer ID co-existing with packet

If a Packet-out message contains both the packet and the buffer ID referencing the packet stored in the switch, the switch only processes the buffered packet and ignores the packet in the message.

Packets without a VLAN tag

If a Packet-out packet has no VLAN tag, the switch tags the packet with the PVID of the ingress port and forwards the packet within the VLAN. The switch processes the packet as follows when the ingress port is a reserved port:

- If the output port is a physical or logical port, the switch tags the packet with the PVID of the output port and forwards the packet within the VLAN.
- If the output port is the Flood or All reserved port, the switch processes the packet as described in "Output port."

Output port

If the output port of a Packet-out packet is the Flood or All reserved port, the switch processes the packet as follows:

- When the output port is the Flood reserved port:
  - o If the Packet-out packet has a VLAN tag, the switch broadcasts the packet within the VLAN.
  - o If the Packet-out packet has no VLAN tag and the ingress port is a physical or logical port, the switch tags the packet with the PVID of the ingress port. The switch then forwards the packet within the VLAN.
  - o If the Packet-out packet has no VLAN tag and the ingress port is the Controller reserved port, the switch forwards the packet out all OpenFlow ports.
- When the output port is the All reserved port:
  - o If the Packet-out packet has a VLAN tag, the switch broadcasts the packet within the VLAN.
  - o If the Packet-out packet has no VLAN tag, the switch forwards the packet out of all OpenFlow ports regardless of the ingress port type.

Packet-in messages restrictions

Processing VLAN tags

When sending a Packet-in packet to the controller, the switch processes the VLAN tag of the packet as follows:

- If the VLAN tag of the packet is the same as the PVID of the ingress port, the switch removes the VLAN tag.
- If the VLAN tag of the packet is different from the PVID of the ingress port, the switch does not remove the VLAN tag.

Packet buffer

If a Packet-in message is sent to controller due to no matching flow entry, the switch supports buffering the packet. The buffer size is 1K packets.

If a Packet-in message is sent to controller for other reasons, the switch does not support buffering the packet. The switch must send the full packet to the controller, and the cookie field of the packet is set to 0xFFFFFFFFFFFFFFFF.

LLDP frame matching

LLDP is used to perform topology discovery in an OpenFlow network. LLDP must be enabled globally on a device. A switch sends a LLDP frame to the controller through the Packet-in message when the following conditions exist:

- The port that receives the LLDP frame from the controller belongs to OpenFlow instances.
- The flow tables in the OpenFlow instance have a flow entry that matches the LLDP frame (the output port is the Controller reserved port).

Flow table modification messages restrictions

The flow table modification messages have the following restrictions for the table-miss flow entry and common flow entries:

- Table-miss flow entry
  - The controller deploys the table-miss flow entry (the action is Drop) to an OpenFlow instance after the OpenFlow instance is activated.
  - The controller cannot query the table-miss flow entry through Multipart messages.
  - The controller cannot modify the table-miss flow entry through the Modify request. The controller can only modify the table-miss flow entry through the Add request.
  - The controller can modify or delete the table-miss flow entry only through the strict version of the Modify or Delete request. The controller cannot modify or remove the table-miss flow entry through the non-strict version of the Modify or Delete request despite that the match fields are wildcarded.
- Common flow entries

  The controller cannot modify or remove all common flow entries through the non-strict version of the Modify or Delete request despite that the match fields are wildcarded.

Appendix B MAC-IP flow table

Support capabilities of the MAC-IP flow table

The controller must include the required match fields and actions and can include the optional match fields and actions in the flow entries deployed to the MAC-IP flow table. If the controller does not include the optional match fields and actions in the flow entries, the switch adds them to the flow entries by default.

The Layer 2 flow entries are implemented by using MAC address tables. Table 2 describes the support capabilities for Layer 2 flow entries.

Table 2 Support capabilities for Layer 2 flow entries

Support items	Capabilities
Required match fields	The MAC-IP flow table must support the following match fields: • VLAN ID. • Unicast destination MAC address.
Optional match fields	N/A
Required actions	Specifying the output port.
Optional actions	The MAC-IP flow table can optionally support the following instructions: • **Goto-Table**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it. • **Write-Metadata**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it.

The Layer 3 flow entries are implemented by using routing tables. Table 3 describes the support capabilities for Layer 3 flow entries.

Table 3 Support capabilities for Layer 3 flow entries

Support items	Capabilities
Required match fields	The MAC-IP flow table must support the following match fields: • VLAN ID. • Unicast destination IP address.
Optional match fields	Unicast destination MAC address. If the controller includes this match field in the flow entries, the destination MAC address must be the VLAN interface of the VLAN that is matched.
Required actions	Specifying the output port.
Optional actions	The MAC-IP flow table can optionally support the following actions: • **Modify source MAC address**—The switch modifies the source MAC address to the MAC address of the virtual interface of the VLAN to which the output port belongs. • **Decrement TTL by one**. • **Goto-Table**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it. • **Write-Metadata**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it.

The VXLAN Layer 2 flow entries are implemented by using MAC address tables. Table 4 describes the support capabilities for VXLAN Layer 2 flow entries.

Table 4 Support capabilities for VXLAN Layer 2 flow entries

Support items	Capabilities
Required match fields	Unicast destination MAC address.
Optional match fields	N/A
Required actions	The MAC-IP flow table must support the following actions: • Specifying a tunnel interface as the output port. • Specifying the tunnel ID.

Support items	Capabilities
Optional actions	The MAC-IP flow table can optionally support the following instructions: • **Goto-Table**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it. • **Write-Metadata**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it.

The VXLAN Layer 3 flow entries are implemented by using routing tables. Table 5 describes the support capabilities for VXLAN Layer 3 flow entries.

Table 5 Support capabilities for VXLAN Layer 3 flow entries

Support items	Capabilities
Required match fields	The MAC-IP flow table must support the following match fields: • Ethernet type. • Unicast destination IP address.
Optional match fields	N/A
Required actions	The MAC-IP flow table must support the following actions: • Specifying a tunnel interface as the output port. • Specifying the tunnel ID. • Modifying the destination address.
Optional actions	The MAC-IP flow table can optionally support the following actions: • **Goto-Table**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it. • **Write-Metadata**—When the switch has multiple tables, the switch adds this instruction by default if the controller does not deploy it.

MAC-IP flow table restrictions

Controller must follow the restrictions in Table 6, Table 7, Table 8, and Table 9 to deploy flow entries for MAC-IP flow table. Otherwise, forwarding failure might occur.

Table 6 Restrictions for deploying Layer 2 flow entries for the MAC-IP flow table

Items	Restrictions
Match fields	The destination MAC address cannot be MAC address of the switch to which the flow entry is deployed.
Actions	The output port must belong to the VLAN that is matched.

Table 7 Restrictions for deploying Layer 3 flow entries for the MAC-IP flow table

Items	Restrictions
Match fields	The VLAN interface of the VLAN that is matched is in UP state. The destination MAC address cannot be MAC address of the switch to which the flow entry is deployed. The destination IP address cannot be the IP address of the switch to which the flow entry is deployed.

Items	Restrictions
Actions	The specified output port must belong to the destination VLAN.
	The destination MAC address cannot be MAC address of the switch to which the flow entry is deployed.
	If the switch modifies the source MAC address, the source MAC address must be the MAC address of the VLAN interface of the VLAN to which the output port belongs.

To deploy a Layer 3 flow entry, make sure the following requirements are met:

- The VLAN interface of the matched VLAN is in up state.
- The switch sends the controller a packet that indicates the VLAN interface functions as an OpenFlow port. The link state and the MAC address of the VLAN interface are also included in the packet.

The switch reports the VLAN interface deletion to the controller and the controller removes the corresponding Layer 3 flow entry.

The controller ensures the correctness of Layer 3 flow entries. The switch does not check for the restrictions for Lay 3 flow entries.

Table 8 Restrictions for deploying VXLAN Layer 2 flow entries for the MAC-IP flow table

Items	Restrictions
Match fields	The destination MAC address cannot be MAC address of the switch to which the flow entry is deployed.
Actions	The output port must be an existing tunnel interface.
	The VXLAN that corresponds to the specified tunnel ID (VNI) must exist.

Table 9 Restrictions for deploying VXLAN Layer 3 flow entries for the MAC-IP flow table

Items	Restrictions
Match fields	The destination IP address cannot be the IP address of the switch to which the flow entry is deployed.
Actions	The output port must be an existing tunnel interface.
	The VXLAN and the VSI interface that correspond to the specified tunnel ID (VNI) must exist.
	The destination MAC address cannot be MAC address of the switch to which the flow entry is deployed.

Table-miss flow entry of MAC-IP flow tables

The table-miss flow entry of a MAC-IP flow table supports the following output actions:

- **Goto-Table**—Direct the packet to the next table.
- **Drop**—Drop the packet.
- **Controller**—Send the packet to the controller.
- **Normal**—Forward the packet to the normal pipeline.

Dynamic aware

On an OpenFlow switch that supports MAC-IP flow tables, you can configure OpenFlow to support querying and deleting dynamic MAC address flow entries.

The controller can query and delete dynamic MAC address flow entries by specifying a VLAN, a MAC address, or the combination of a MAC address and a VLAN.

MAC-IP flow table cooperating with extensibility flow table

The MAC-IP flow table supports the Write MetaData/Mask instruction and the extensibility flow table supports metadata/mask matching. The MAC-IP flow table can cooperate with the extensibility flow table to perform the pipeline process of multiple tables by using metadata/mask.

Each metadata mask bit has a different meaning. The corresponding metadata bit being set indicates that the metadata mask bit is matched. When the corresponding metadata bit is not set, the metadata mask bit is wildcarded.

Table 10 Metadata mask meanings

Metadata mask bit	Meaning	Metadata
Bit 0	Destination MAC address	• **1**—Set. Matches the destination MAC address. • **0**—Not set. Does not match the destination MAC address.
Bit 1	Source MAC address	• **1**—Set. Matches the source MAC address. • **0**—Not set. Does not match the source MAC address.
Bit 2	Destination IP address	• **1**—Set. Matches the destination IP address. • **0**—Not set. Does not match the destination IP address.
Others	Reserved	Reserved.

Document conventions and icons

Conventions

This section describes the conventions used in the documentation.

Port numbering in examples

The port numbers in this document are for illustration only and might be unavailable on your device.

Command conventions

Convention	Description
Boldface	**Bold** text represents commands and keywords that you enter literally as shown.
Italic	*Italic* text represents arguments that you replace with actual values.
[ ]	Square brackets enclose syntax choices (keywords or arguments) that are optional.
{ x | y | ... }	Braces enclose a set of required syntax choices separated by vertical bars, from which you select one.
[x | y | ...]	Square brackets enclose a set of optional syntax choices separated by vertical bars, from which you select one or none.
{ x | y | ... } *	Asterisk marked braces enclose a set of required syntax choices separated by vertical bars, from which you select at least one.
[x | y | ...] *	Asterisk marked square brackets enclose optional syntax choices separated by vertical bars, from which you select one choice, multiple choices, or none.
&<1-n>	The argument or keyword and argument combination before the ampersand (&) sign can be entered 1 to n times.
#	A line that starts with a pound (#) sign is comments.

GUI conventions

Convention	Description
Boldface	Window names, button names, field names, and menu items are in Boldface. For example, the **New User** window appears; click **OK**.
>	Multi-level menus are separated by angle brackets. For example, **File** > **Create** > **Folder**.

Symbols

Convention	Description
⚠ WARNING!	An alert that calls attention to important information that if not understood or followed can result in personal injury.
△ CAUTION:	An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software.
ⓘ IMPORTANT:	An alert that calls attention to essential information.
NOTE:	An alert that contains additional or supplementary information.
◠ TIP:	An alert that provides helpful information.

Network topology icons

Convention	Description
	Represents a generic network device, such as a router, switch, or firewall.
	Represents a routing-capable device, such as a router or Layer 3 switch.
	Represents a generic switch, such as a Layer 2 or Layer 3 switch, or a router that supports Layer 2 forwarding and other Layer 2 features.
	Represents an access controller, a unified wired-WLAN module, or the access controller engine on a unified wired-WLAN switch.
	Represents an access point.
	Represents a wireless terminator unit.
	Represents a wireless terminator.
	Represents a mesh access point.
	Represents omnidirectional signals.
	Represents directional signals.
	Represents a security product, such as a firewall, UTM, multiservice security gateway, or load balancing device.
	Represents a security card, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG card.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  www.hpe.com/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  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, go to either of the following:

  o Hewlett Packard Enterprise Support Center **Get connected with updates** page:

    www.hpe.com/support/e-updates

  o Software Depot website:

    www.hpe.com/support/softwaredepot

- To view and update your entitlements, and to link your contracts, Care Packs, 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 HP Passport set up with relevant entitlements.

Websites

Website	Link
Networking websites	
Hewlett Packard Enterprise Information Library for Networking	www.hpe.com/networking/resourcefinder
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/enterprise/docs
Hewlett Packard Enterprise Support Center	www.hpe.com/support/hpesc
Hewlett Packard Enterprise Support Services Central	ssc.hpe.com/portal/site/ssc/
Contact Hewlett Packard Enterprise Worldwide	www.hpe.com/assistance
Subscription Service/Support Alerts	www.hpe.com/support/e-updates
Software Depot	www.hpe.com/support/softwaredepot
Customer Self Repair (not applicable to all devices)	www.hpe.com/support/selfrepair
Insight Remote Support (not applicable to all devices)	www.hpe.com/info/insightremotesupport/docs

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:

www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty, Care Pack Service, 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.

For more information and device support details, go to the following website:

www.hpe.com/info/insightremotesupport/docs

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.

Index

31