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.

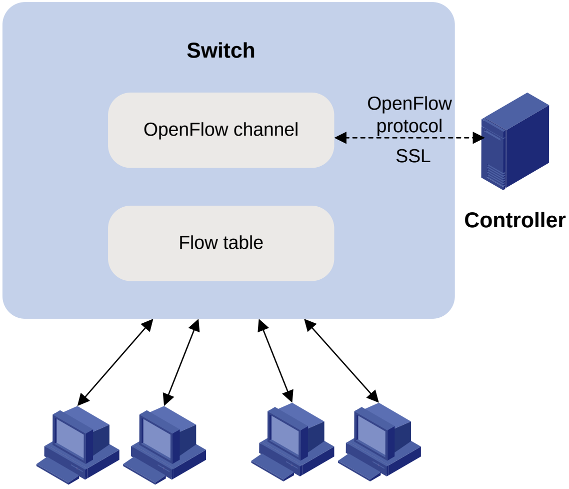
Overview

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 operations:

- 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 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.
    - **Local**—Local CPU.
    - **Normal**—Normal forwarding process.
    - **Flood**—Flooding.

  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

An OpenFlow instance operates in one of the following modes:

- **Global mode**—When the global mode is enabled for an OpenFlow instance, the flow entries take effect on packets within the network.
- **VLAN mode**—When the VLAN mode is enabled for an OpenFlow instance, the flow entries take effect only on packets within VLANs associated with 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:

- Capabilities supported by OpenFlow.
- Information about ports that belong 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.

In loosen mode, a port belongs to the OpenFlow instance when VLANs associated with the OpenFlow instance overlap with the port's allowed VLANs. Otherwise, a port belongs to an OpenFlow instance only when VLANs associated with the OpenFlow instance are within the port's allowed VLAN list.

OpenFlow flow table

An OpenFlow switch matches packets with 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.

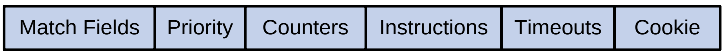
  A MAC-IP flow table provides the following match fields:
  - Destination MAC address.
  - VLAN.
  - Destination IP address.

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

  For more information, see "Appendix B MAC-IP flow table."
- **Extensibility**—Uses ACLs to match packets.
- **VLAN tagging**—Tags all incoming packets matching the table.
- **VLAN untagging**—Untags all outgoing packets matching the table.

Flow entry

Figure 2 Flow entry components



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 with 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 multiple flow tables exist.
  - **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**—No explicit action exists 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 the output action, the packet is assigned to the queue attached to this port for scheduling and forwarding. The forwarding behavior is dictated by the configuration of the queue and provides basic QoS support.
  - **(Optional.) Push-Tag/Pop-Tag**—Switches support the ability to push or pop tags, such as VLAN 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.
- **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 with 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

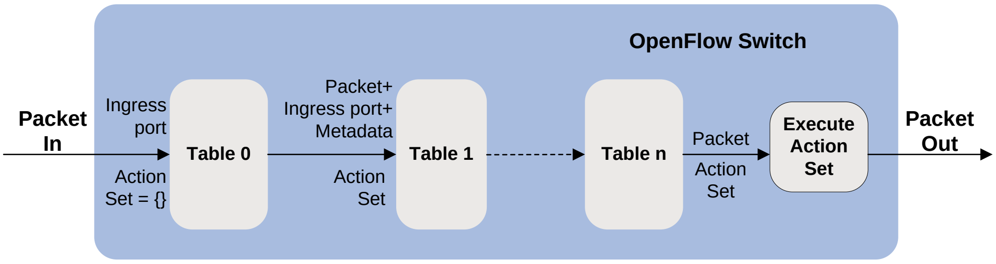


Table-miss flow entry

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.

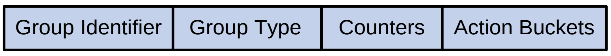
The table-miss flow entry wildcards all match fields (all fields omitted) and has the lowest priority 0.

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



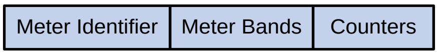
A group entry contains the following fields:

- **Group Identifier**—A 32 bit unsigned integer uniquely identifying the group.
- **Group Type**—Type of the group. **All** indicates that all buckets in the group are executed. This group is used for multicast or broadcast forwarding.
- **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.

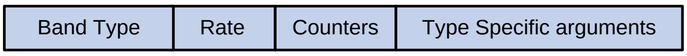
Figure 5 Meter entry components



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



A meter band contains the following fields:

- **Band Type**—(Optional.) Packet processing methods. Options are:
  - **Drop**—Discards the packet when the rate of the packet exceeds the band rate.
  - **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 with the switch to perform the following operations:

- Configure and manage the switch.
- Receive events from the switch.
- 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 representing 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.

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: ○ (Required.) Configuring the OpenFlow instance mode ○ (Optional.) Configuring flow tables for an OpenFlow instance ○ (Optional.) Setting the controller mode ○ (Optional.) Setting the maximum number of flow entries for the extensibility flow table ○ (Optional.) Configuring inband management VLANs ○ (Optional.) Configuring OpenFlow to forbid MAC address learning ○ (Optional.) Setting the datapath ID ○ (Optional.) Enabling an SSL server for an OpenFlow instance ○ (Optional.) Configuring the default action of table-miss flow entries ○ (Optional.) Preventing an OpenFlow instance from reporting the specified types of ports to controllers **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
(Optional.) Excluding the specified VLANs from the VLANs in which traffic is forwarded in the OpenFlow forwarding process
(Optional.) Setting OpenFlow timers
(Optional.) Configuring an OpenFlow instance to support dynamic MAC addresses
(Optional.) Creating a highest-priority flow entry for dropping slow protocol packets
(Optional.) Allowing dynamic ARP entries to overwrite OpenFlow ARP entries
(Optional.) Enabling an OpenFlow instance to perform QinQ tagging for double-tagged packets passing an

Tasks at a glance
extensibility flow table
(Optional.) Setting a DSCP value for OpenFlow packets
(Optional.) Disabling logging for successful flow table modifications
(Optional.) Refreshing all Layer 3 flow entries in the MAC-IP flow tables for an OpenFlow instance
(Optional.) Allowing link aggregation member ports to be in the deployed flow tables
(Optional.) Preventing an OpenFlow instance from reporting ARP packets to the specified controllers
(Optional.) Enabling OpenFlow connection backup
(Optional.) Shutting down an interface by OpenFlow
(Optional.) Enabling loop guard for an OpenFlow instance

Configuring OpenFlow instances

Creating an OpenFlow instance

To configure an OpenFlow instance:

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

Configuring the OpenFlow instance mode

When you associate an OpenFlow instance with VLANs, follow these guidelines:

- For VLAN traffic to be processed correctly, do not associate multiple OpenFlow instances with the same VLAN.
- When you activate an OpenFlow instance that is associated with non-existent VLANs, the system automatically creates the VLANs.
- Do not configure BFD MAD on the VLAN interface for a VLAN that is associated with an OpenFlow instance. For more information about BFD MAD, see *Virtual Technologies Configuration Guide*.

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** | **vlan** *vlan-id* [**mask** *vlan-mask*] [**loosen**] }	By default, the OpenFlow instance mode is not configured.

Configuring flow tables for an OpenFlow instance

If you specify the **ingress-vlan** *ingress-table-id* option, make sure the VLAN tagging flow table has the smallest ID among all flow tables. If you specify the **egress-vlan** *egress-table-id* option, make sure the VLAN untagging flow table has the largest ID among all flow tables. The VLAN tagging flow table and untagging flow table take effect only when the following conditions are met:

- The OpenFlow instance is configured to perform QinQ tagging for double-tagged packets passing an extensibility flow table.
- The device operates in standalone mode.

To configure 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. Configure flow tables for the OpenFlow instance.	**flow-table** { [**ingress-vlan** *ingress-table-id*] [**extensibility** *extensibility-table-id* | **mac-ip** *mac-ip-table-id*] * [**egress-vlan** *egress-table-id*] }	By default, an OpenFlow instance contains one extensibility flow table with an ID of 0.

Setting the controller mode

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 the extensibility flow table

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

To set the maximum number of flow entries that the extensibility flow table supports:

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 maximum number of flow entries that the extensibility flow table supports.	**flow-entry max-limit** *limit-value*	By default, the extensibility flow table can have a maximum of 65535 flow entries.

Configuring inband management VLANs

You can configure inband management VLANs for an OpenFlow instance. Traffic in these VLANs is forwarded in the normal forwarding process instead of the OpenFlow forwarding process. The ports that are assigned only to inband management VLANs are not OpenFlow ports.

To configure inband management VLANs:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Configure inband management VLANs.	**in-band management vlan** { *vlan-id* [**to** *vlan-id*] } &<1-10>	By default, no inband management VLAN is configured for an OpenFlow instance. Inband management VLANs that you configure for an OpenFlow instance must be within the list of the VLANs that are associated with the OpenFlow instance.

Configuring OpenFlow to forbid MAC address learning

You can configure this feature for an OpenFlow instance to forbid MAC address learning for VLANs associated with the OpenFlow instance. The configuration does not take effect on inband management VLANs.

To forbid MAC address learning for VLANs associated with 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. Forbid MAC address learning for VLANs associated with the OpenFlow instance.	**mac-learning forbidden**	By default, MAC address learning is allowed for VLANs associated with an OpenFlow instance.

Setting the datapath ID

The datapath ID uniquely identifies an OpenFlow switch (OpenFlow instance). Do not set the same datapath ID for different OpenFlow switches.

To set the datapath ID:

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 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.

For more information about SSL, see *Security Configuration Guide*.

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.	**openflow instance** *instance-id*	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. To re-configure the SSL server, first execute the **undo** form of the command to delete the existing SSL server configuration.

Configuring the default action of table-miss flow entries

Packets that do not match the MAC-IP flow entries are matched with ARP or MAC entries before table-miss flow entries. If a match is found, the packets are forwarded in the normal forwarding process. If no match is found, the packets are dropped.

To configure the default action of table-miss flow entries:

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 default action of table-miss flow entries to forward packets to the	**default table-miss permit**	By default, the default action of table-miss flow entries is to drop

Step	Command	Remarks
normal pipeline.		packets.

Preventing an OpenFlow instance from reporting the specified types of ports to controllers

Perform this task to prevent an OpenFlow instance from reporting controllers information about the following types of interfaces that belong to the OpenFlow instance:

- Layer 3 Ethernet interface.
- Layer 3 aggregate interface.
- VLAN interface.

To prevent an OpenFlow instance from reporting 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. Prevent an OpenFlow instance from reporting the specified types of ports to controllers.	**forbidden port** { **l3-physical-interface** | **vlan-interface** } *	By default, all ports that belong to an OpenFlow instance are reported to the controllers.

Activating or reactivating 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. Activate or reactivate the OpenFlow instance.	**active instance**	By default, an OpenFlow instance is not activated.

Configuring controllers for an OpenFlow switch

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 operations:

  - 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 operations 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 configure the main connection to the controller.	**controller** *controller-id* **address** { **ip** *ipv4-address* | **ipv6** *ipv6-address* } [**port** *port-number*] [**local address** { **ip** *local-ipv4-address* | **ipv6** *local-ipv6-address* } [**port** *local-port- number*]] [**ssl** *ssl-policy-name*] [**vrf** *vrf-name*]	By default, an OpenFlow instance does not have a main connection to a controller. 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

Step	Command	Remarks
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** *ipv4-address* | **ipv6** *ipv6-address* }] [**port** *port-number*]	By default, an OpenFlow instance does not have auxiliary connections to a controller.

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. If the output action in a matching flow entry is to forward traffic to a controller, the traffic is discarded.
- **Smart**—The OpenFlow switch forwards traffic based on flow tables and does not remove unexpired flow entries. If the output action in a matching flow entry is to forward traffic to a controller, the traffic is forwarded in normal process.
- **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** | **smart** | **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.

Excluding the specified VLANs from the VLANs in which traffic is forwarded in the OpenFlow forwarding process

The VLANs to be associated with an OpenFlow instance are calculated by a bitwise AND operation on the VLAN ID and mask. Use this feature to exclude the VLANs in which traffic is required to be forwarded in the normal forwarding process from the calculated VLANs. Traffic in the excluded VLANs is forwarded in the normal forwarding process instead of the OpenFlow forwarding process.

To exclude the specified VLANs from the VLANs in which traffic is forwarded in the OpenFlow forwarding process:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Exclude the specified VLANs from the VLANs in which traffic is forwarded in the OpenFlow forwarding process.	**openflow normal-forward vlan** { *vlan-id* [**to** *vlan-id*] } &<1-10>	By default, no VLANs are excluded from the VLANs in which traffic is forwarded in the OpenFlow forwarding process..

Setting OpenFlow timers

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.	**openflow instance** *instance-id*	N/A
3. Set the echo request interval.	**controller echo-request interval** *interval*	The default setting is 5 seconds.
4. Set the interval for the OpenFlow instance to reconnect to a controller.	**controller connect interval** *interval*	The default setting is 60 seconds.

Configuring an OpenFlow instance to support dynamic MAC addresses

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 an OpenFlow instance 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*	N/A
3. Configure the OpenFlow instance to support dynamic MAC addresses.	**mac-ip dynamic-mac aware**	By default, an OpenFlow instance does not support dynamic MAC addresses. An OpenFlow instance ignores dynamic MAC address messages sent from controllers.

Creating a highest-priority flow entry for dropping slow protocol packets

Perform this task to create a highest-priority flow entry for dropping slow protocol (such as LACP, LAMP, and OAM) packets. This entry has a higher priority than the flow entries deployed by controllers.

To create a highest-priority flow entry for dropping slow protocol packets:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Create a highest-priority flow entry for dropping slow protocol packets.	**protocol-packet filter slow**	By default, an OpenFlow instance does not have a highest-priority flow entry for dropping slow protocol packets.

Allowing dynamic ARP entries to overwrite OpenFlow ARP entries

Perform this task to increase the precedence of dynamic ARP entries to overwrite OpenFlow ARP entries. OpenFlow ARP entries are generated based only on the MAC-IP flow table of an OpenFlow instance.

To allow dynamic ARP entries to overwrite OpenFlow ARP entries:

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 to allow dynamic ARP entries to overwrite OpenFlow ARP entries.	**precedence dynamic arp**	By default, an OpenFlow instance does not allow dynamic ARP entries to overwrite OpenFlow ARP entries.

Enabling an OpenFlow instance to perform QinQ tagging for double-tagged packets passing an extensibility flow table

By default, a double-tagged packet becomes single-tagged after it passes an extensibility flow table. Perform this task to allow double-tagged packets to keep double-tagged after the packets pass an extensibility flow table.

To enable an OpenFlow instance to perform QinQ tagging for double-tagged packets passing an extensibility flow table:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Enable the OpenFlow instance to perform QinQ tagging for double-tagged packets passing an extensibility flow table.	**qinq-network enable**	By default, a double-tagged packet becomes single-tagged after it passes an extensibility flow table.

Setting a DSCP value for OpenFlow packets

Step	Command	Remarks
1. Enter system view	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Set a DSCP value for OpenFlow packets.	**tcp dscp** *dscp-value*	By default, the DSCP value for OpenFlow packets is not set. This configuration takes effect only on OpenFlow packets over the main connection that the OpenFlow instance establishes with a controller through TCP.

Disabling logging for successful flow table modifications

This feature disables logging for successful flow table modifications. Logging for other events is not affected.

To disable logging for successful flow table modifications:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Disable logging for successful flow table modifications.	**flow-log disable**	By default, logging for successful flow table modifications is enabled.

Refreshing all Layer 3 flow entries in the MAC-IP flow tables for an OpenFlow instance

Perform this task to obtain all Layer 3 flow entries in the MAC-IP flow tables from the controller again if the Layer 3 flow entries have been overwritten.

To refresh all Layer 3 flow entries in the MAC-IP flow tables for an OpenFlow instance:

Step	Command
1. Enter system view	**system-view**
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*
3. Refresh all Layer 3 flow entries in the MAC-IP flow tables.	**refresh ip-flow**

Allowing link aggregation member ports to be in the deployed flow tables

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A
3. Allow link aggregation member ports to be in the deployed flow tables.	**permit-port-type member-port**	By default, link aggregation member ports cannot be in the deployed flow tables.

Preventing an OpenFlow instance from reporting ARP packets to the specified controllers

This feature forbids an OpenFlow instance to report ARP packets to the specified controllers to prevent the controllers from being affected by a large number of packets.

To prevent an OpenFlow instance from reporting ARP packets to the specified controllers:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view	**openflow instance** instance-id	N/A
3. Prevent the OpenFlow instance from reporting ARP packets to the specified controllers.	**forbidden packet-in arp controller** *controller-id-list*	By default, no controllers to which ARP packets are forbidden to be reported are configured.

Enabling OpenFlow connection backup

This feature enables an OpenFlow instance to back up OpenFlow connections established over TCP. This prevents connection interruption when an active/standby switchover occurs.

To enable OpenFlow connection backup:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter OpenFlow instance view.	**openflow instance** *instance-id*	N/A

Step	Command	Remarks
3. Enable OpenFlow connection backup.	**tcp-connection backup**	By default, OpenFlow connection backup is enabled.

Shutting down an interface by OpenFlow

After an interface is shut down by OpenFlow, the **Current state** field displays **OFP DOWN** in the **display interface** command output. You can use the **undo openflow shutdown** command to bring up an interface shut down by OpenFlow. The interface can also be brought up by port modification messages from controllers.

To shut down an interface by OpenFlow:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Shut down an interface by OpenFlow.	**openflow shutdown**	By default, an interface is not shut down by OpenFlow.

Enabling loop guard for an OpenFlow instance

After an OpenFlow instance is deactivated, loops might occur in VLANs associated with the OpenFlow instance. To avoid loops, you can enable loop guard for the OpenFlow instance. This feature enables the deactivated OpenFlow instance to create a flow entry for dropping all traffic in theses VLANs.

To enable loop guard 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. Enable loop guard for the OpenFlow instance.	**loop-protection enable**	By default, loop guard is disabled for an OpenFlow instance.

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*]
Display flow table entries for an OpenFlow instance.	**display openflow instance** *instance-id* **flow-table** [*table-id*]
Display controller information for an OpenFlow instance.	**display openflow instance** *instance-id* { **controller** [*controller-id*] | **listened** }
Display group information for an OpenFlow instance.	**display openflow instance** *instance-id* **group**

Task	Command
	[*group-id*]
Display meter information for an OpenFlow instance.	**display openflow instance** *instance-id* **meter** [*meter-id*]
Display summary OpenFlow instance information.	**display openflow summary**
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** } **statistics**

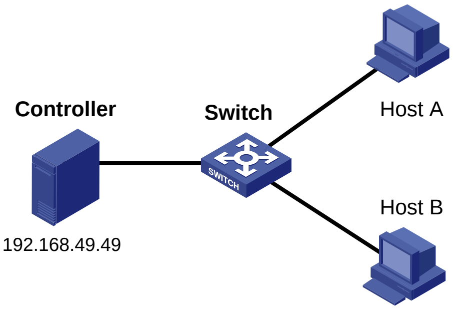
OpenFlow configuration example

Network requirements

As shown in Figure 7, an OpenFlow switch communicates with the controller. Perform the following tasks on the OpenFlow switch:

- Create OpenFlow instance 1, associate VLANs 4092 and 4094 with the OpenFlow instance, and activate the OpenFlow instance.
- Configure the IP address for controller 1 to have the controller manage the OpenFlow switch.

Figure 7 Network diagram



Configuration procedure

# Create VLANs 4092 and 4094.

```
<Switch> system-view
[Switch] vlan 4092
[Switch-vlan4092] quit
[Switch] vlan 4094
[Switch-vlan4094] quit
```

# Create OpenFlow instance 1 and associate VLANs with it.

```
[Switch] openflow instance 1
[Switch-of-inst-1] classification vlan 4092 mask 4093
```

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

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

Verifying the configuration

# View detailed information about the OpenFlow instance.

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

Configuration information:
 Description   : --
 Active status : Active
 Inactive configuration:
  None
 Active configuration:
  Classification VLAN, total VLANs(2)
   4092, 4094
  In-band management VLAN, total VLANs(0)
   Empty VLAN
  Connect mode: Multiple
  Mac-address learning: Enabled
  TCP DSCP value: 10
  Flow table:
   Table ID(type): 0(Extensibility), count: 0
  Flow-entry max-limit: 65535
  Datapath ID: 0x0064001122000101
  Default table-miss: Drop
  Forbidden port: None
  Qinq Network: Disabled
  TCP connection backup: Enabled
Port information:
 Ten-GigabitEthernet1/0/3
Active channel information:
 Controller 1 IP address: 192.168.49.49 port: 6633
```

Appendixes

Appendix A Application restrictions

Matching restrictions

VLAN matching

Table 1 describes the VLAN matching restrictions when an OpenFlow instance is associated with VLANs.

Table 1 VLAN matching

VLAN	Mask	Matching packets
-	-	All packets in the VLANs that are associated the OpenFlow instance.
0	-	Packets without a VLAN tag. The PVID of the ingress port must be associated the OpenFlow instance.
0	Value	Unsupported.
Valid VLAN	-/value	Unsupported.
0x1000	-/value(except non-0x1000)	Unsupported.
0x1000	0x1000	Packets with a VLAN tag. The VLAN ID of the VLAN tag must be associated with the OpenFlow instance.
Valid VLAN | 0x1000	-/value	Matching packets by the combination of the VLAN ID and VLAN mask. The VLANs obtained through the combination of the VLAN ID and VLAN mask must be associated with the OpenFlow instance.
Other	Other	Unsupported.

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 passes 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 2 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

Instruction type	Restrictions
	time.
Apply-Actions	The action list of the Apply-Actions instruction cannot include multiple Output actions. When the action list 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, the following rules apply:
  - If actions in the action set do not conflict with actions in the action list, the switch merges the action list into the action set.
  - If actions in the action set conflict with actions in the action list, actions in the action list are replaced with actions in the action set.
- When the action set and the action list contain the Output action or the Group action, the following rules apply:
  - If both the action list and the action set contain an Output action, the Output action in the action list takes precedence. The Output action in 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.
  - 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. The actions are executed in the order defined by the action set rules.
  - If the action list contains an Output action and the action set contains a Group action, the following rules apply:
    - The Output action does not modify the packet.
    - The Group action is executed.

Packet-out messages restrictions

Ingress port

The ingress port must be a physical or logical port when one of the following reserved ports is the output port in a packet-out message:

- Normal.
- Local.
- In Port.
- Controller.

Buffer ID co-existing with packet

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

Packets without a VLAN tag

If the packet contained in a packet-out message has no VLAN tag, the switch performs the following operations:

- Tags the packet with the PVID of the ingress port.
- 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 in a packet-out message is the Flood or All reserved port, the switch processes the packet contained in the packet-out message as follows:

- When the output port is the Flood reserved port:
  - If the packet has a VLAN tag, the switch broadcasts the packet within the VLAN.
  - If the 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.
  - If the 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:
  - If the packet has a VLAN tag, the switch broadcasts the packet within the VLAN.
  - If the 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 message to the controller, the switch processes the VLAN tag of the packet contained in the packet-out message 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 contained in the packet-in message. 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 contained in the packet-in message. 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.
  - The controller deploys a table-miss flow entry (the action is Drop) to an OpenFlow instance after the current table-miss flow entry is deleted.
- 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.

Mirroring restrictions

The device does not support configuring Layer 3 remote port mirroring through OpenFlow.

Appendix B MAC-IP flow table

Capabilities supported by 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 entries. Table 3 describes the capabilities supported by Layer 2 flow entries.

Table 3 Capabilities supported by Layer 2 flow entries

Item	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.

Item	Capabilities
	• **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 entries. Table 4 describes the capabilities supported by Layer 3 flow entries.

Table 4 Capabilities supported by Layer 3 flow entries

Item	Capabilities
Required match fields	The MAC-IP flow table must support the following match fields: • VLAN ID. • Unicast destination IP address. • Unicast destination MAC address, which must be the MAC address of the VLAN interface for the VLAN that is matched.
Optional match fields	N/A
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 VLAN interface for 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.

MAC-IP flow table restrictions

Controller must follow the restrictions in Table 5 and Table 6 to deploy flow entries for MAC-IP flow table. Otherwise, forwarding failure might occur.

Table 5 Restrictions for deploying Layer 2 flow entries for the MAC-IP flow table

Items	Restrictions
Match fields	The destination MAC address cannot be the 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 6 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 is the MAC address of the VLAN interface for the VLAN that is matched. The destination IP address cannot be the IP address of the switch to which the flow entry is deployed.
Actions	The specified output port must belong to the destination VLAN. The destination MAC address cannot be the 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 for the VLAN to which the output port

Items	Restrictions
	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 acts 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-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

Metadata/mask

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 an 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 7 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.

Metadata mask bit	Meaning	Metadata
Others	Reserved	Reserved.

Matching restrictions

When the output action in an extensibility flow table is not Normal, the following rules apply:

- The MAC-IP flow table does not take effect.
- All actions are executed according to the extensibility flow table.

When the output action in an extensibility flow table is Normal, the following rules apply:

- The output action is executed according to the MAC-IP flow table.
- The other actions are executed according to the extensibility flow table.

Appendix C VLAN tagging and untagging flow tables

Capabilities supported by the VLAN tagging 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 VLAN tagging 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.

Table 8 describes the capabilities supported by the flow entries in the VLAN tagging flow table.

Table 8 Capabilities supported by flow entries in the VLAN tagging flow table

Item	Capabilities
Required match fields	The VLAN tagging flow table must support the following match fields: • **input-port**. • **vlan**.
Optional match fields	N/A
Required actions	The following actions in the action list of the Apply-Actions instruction must be applied immediately: • **Push-Tag**. • **Set-Field (vlan)**.
Optional actions	The VLAN tagging flow table can optionally support the following actions: • **Output (normal)**. • **Goto-Table**.

The Push-Tag and Set-Field (vlan) actions must be in the action list of the Apply-Actions instruction. The Push-Tag and Set-Field (vlan) actions can be used as follows:

- **Push-Tag + Set-Field (value1)**—Adds a VLAN tag **value1**.
- **Set-Field (value1) + Push-Tag + Set-Field (value2)**—Modifies the VLAN tag of the packet to **value1** and adds a VLAN tag **value2**.
- **Push-Tag + Set-Field (value1) + Push-Tag + Set-Field (value2)**—Adds inner VLAN tag **value2** and outer VLAN tag **value1**.

The **Goto-Table** instruction is optional and does not take effect. The flow table specified by this instruction can only be the next of the VLAN tagging flow table.

Capabilities supported by the VLAN untagging flow table

Table 9 Capabilities supported by flow entries in the VLAN untagging flow table

Item	Capabilities
Required match fields	The VLAN untagging flow table must support the following match fields: • **egress port**—Matches the egress port of packets. • **vlan**—Matches the outer VLAN tag of packets.
Optional match fields	The VLAN untagging flow table can optionally support the **inner vlan** match field that matches the inner VLAN tag of double-tagged packets.
Required actions	The following actions in the action list of the Apply-Actions instruction must be applied immediately: • **Pop-Tag**. • **Set-Field (vlan)**.
Optional actions	The VLAN untagging flow table can optionally support the **Output (normal)** action.

The VLAN untagging flow table applies only to double-tagged packets.

The **egress port** and **inner vlan** are extended match fields that use the Experimenter ID (0xFE2) and take the private match field values 47 and 48, respectively. To deploy flow entries that contain the extended match fields, make sure the controllers are developed to be compatible with the Experimenter ID and the extended match fields.

The Pop-Tag and Set-Field (vlan) actions must be in the action list of the Apply-Actions instruction. The Pop-Tag and Set-Field (vlan) actions can be used as follows:

- **Pop-Tag**—Removes the outer VLAN tag.
- **Pop-Tag + Pop-Tag**—Removes the inner and outer VLAN tags.
- **Pop-Tag + Set-Field (value)**—Removes the outer VLAN tag and modifies the inner VLAN tag to **value**.