Contents

1 Configuring OpenFlow

1.1 Introduction

1.1.1 Overview

OpenFlow is a network transmission protocol defined between a controller and the forwarding layer in the software defined network (SDN) architecture. OpenFlow allows the controller to directly access and operate network devices on the forwarding plane. These devices may be physical or virtualized.

OpenFlow separates the control plane from the forwarding plane of the network devices through standard communication protocols. Thus, the network devices focus on forwarding, and the control of an entire network focuses one controller. An OpenFlow connection is established between a controller and a device. The controller generates and delivers forwarding rules to the network device in a flow table through the OpenFlow connection, thereby centrally manage the control plane and reducing maintenance and management costs.

1.1.2 Principles

1. **Basic Concepts**

- Flow table

  A flow table is a core data structure for a device to control forwarding policies. Based on the flow table, the device determines an action to be taken for network traffic that enters the device.

  A device can have one or multiple flow tables. At present, a device supports only one flow table. A flow table is composed of multiple flow entries.

  According to the OpenFlow protocol, a flow entry consists of three parts: **header**, **counter**, and **action**.

  - **header**: It defines the index of a flow entry and consists of various packet fields such as source Media Access Control (MAC) address, destination MAC address, Ethernet protocol type, source IP address, destination IP address, IP protocol type, source port, and destination port. If a packet matches the header field of a flow entry, the packet matches this flow entry.
  - **counter**: It counts the number of packets and bytes that match this flow entry.
  - **action**: It defines an action taken for the packet that matches the flow entry, for example, discarding, broadcasting, or forwarding the packet.

- Message

  The OpenFlow protocol supports three categories of messages: **controller-to-switch**, **asynchronous**, and **symmetric**. Each category of messages include several subtypes of messages. The three categories of messages are described as follows:

  - **controller-to-switch**: This category of messages are initiated by the controller to manage and obtain network device statuses. For the subtypes of messages, see Table 1-1.
  - **asynchronous**: This category of messages are initiated by a network device to update network events or network device status changes (most commonly link up/down of a network port) to the controller. For the

subtypes of messages, see Table 1-2.

- **symmetric**: This category of messages are initiated either by a device or a controller to achieve an initial handshake of the protocol and detect connection statuses. For the subtypes of messages, see Table 1-3.

Table 1-1Subtypes of controller-to-switch Messages

Subtype	Description
Features	Used by a controller to send Features requests to understand device capabilities and responses.
Configuration	Used by a controller to query device configuration and make responses.
Modify-State	Used to manage device statuses, for example, flow entry and port statuses. This type of messages are used to add, delete, and modify flow entries, group entries, and port attributes of a device.
Multipart	Used by a controller to request device information, for example, current configuration and statistics.
Packet-Out	Used by a controller to send packets to a device.
Barrier	Used to check whether an action taken before a barrier message is successful. A controller sends a barrier request message. If a device confirms that the preceding action is successful, the device returns a barrier response message.
Role-Request	Used by a network device to request the role of a controller.
Asynchronous-Configuration	Used by a controller to set an asynchronous message filter and filter out unwanted asynchronous messages or query this filter.

Table 1-2Subtypes of Asynchronous Messages

Subtype	Description
Packet-in	Used by a device to send packets to the controller.
Flow-Removed	Used to notify a controller of removing a flow entry from the flow table.
Port-Status	Used to notify a controller of the change in the device port status or configuration.
Error	Used to notify a controller of a device fault or error.

Table 1-3Subtypes of Symmetric Messages

Subtype	Description
Hello	A message for interaction sent by between a controller and a device when

Subtype	Description
	an OpenFlow connection is established.
Echo	Keep-alive (echo request/reply) messages sent by a controller and a device.
Experimenter	A message reserved for new features.

2. **Roles of the Controller**

Table 1-1Roles of the Controller

Role	Permission
Master	The controller in this role has all the permissions to control devices, deliver a flow table, query statistics, and receive status messages reported by the devices. In the cluster deployment mode, only one controller plays the master role.
Equal	The controller in this role has all the permissions of the controller. Compared with the master role, the equal role can be played by multiple controllers.
Standby	The controller in this role has partial permissions, and it is not permitted to deliver flow entries, group entries, or meter entries. The controller is not permitted to modify device configuration or perform packet out operation.

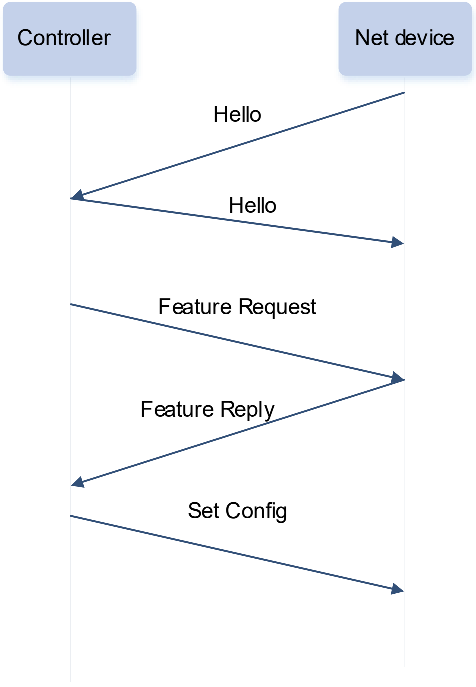
3. **Separating Control from Forwarding**

OpenFlow enables centralized management of the network control plane, so that the entire network can be centrally managed at ease (as compared with the status quo of the network), reducing maintenance and management costs.

The controller sends flow table information to the network devices, so as to control the method for forwarding network data packets and some configuration parameters. Each network device will send a notification message to the controller when its link is interrupted or when the network device receives a data packet in which no forwarding action has been specified. In this way, the interaction between the controller and the network devices is implemented to eventually control the transmission of the entire network.

The mutual discovery between a controller and a network device must be completed before the controller and the network device interact with each other. Figure 1-1 shows the specific actions involved in this process.

Figure 1-1Principle of the Controller and Network Device Interaction



Hello packets are sent between a controller and a network device to achieve a handshake. After the handshake is completed, the controller requests specific information about the device, including the number of ports on the network device and the capability of each port (such as the Feature Request/Reply shown in Figure 1-1). Then the controller delivers specific user configurations (such as Set Config shown in Figure 1-1) to the network device. After a connection is established, the controller defines various flows and corresponding actions for the flows, and delivers them in a flow table to the network device. When a data packet enters the network device, the network device matches the data packet with the flow table according to present flow table rules and performs a corresponding action (including forwarding, discarding, and modifying the packet). At the same time, a corresponding counter is updated. If no match is found in the flow table, the network device forwards the data packet to the controller.

The network device locally maintains the flow table delivered from the controller. If the data packet to be forwarded is already defined in the flow table, the network device directly forwards the data packet. Otherwise, the data packet is sent to the controller to confirm the transmission path (which can be understood as control plane parsing to generate the flow table) and then forwarded based on the flow table delivered from the controller.

1.1.3 Protocols and Standards

- OpenFlow Switch Specification Version 1.0.0
- OpenFlow Switch Specification Version 1.3.0

1.2 Restrictions and Guidelines

- Only the Table 0 flow table is supported.
- A maximum of three controllers can be connected.

1.3 Configuration Task Summary

OpenFlow configuration includes the following tasks:

(1) Configuring OpenFlow

(2) (Optional) Configuring the OpenFlow Multi-controller Mode

(3) (Optional) Configuring the VLAN Tag for Packets Sent to the Controller

(4) (Optional) Configuring the Table-Lookup Mode

(5) (Optional) Configuring the OpenFlowSource IP Address

1.4 Configuring OpenFlow

1.4.1 Overview

As a network transmission protocol, OpenFlow runs over Transport Layer Security (TLS) or unprotected TCP connections, and defines the interaction between a controller and a network device.

1.4.2 Configuring OpenFlow

1. **Overview**

After the OpenFlow function is enabled on a device, packet forwarding control policies are generated and delivered by a controller.

2. **Restrictions and Guidelines**

- Before switching the address of a controller, disable and then enable the OpenFlow function again.
- When the OpenFlow function is enabled, the LLDP function on the device must be disabled. Otherwise, the LLDP packets delivered by a controller cannot be sent out.

3. **Procedure**

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the OpenFlow function.

of controller-ip *ipv4-address* [**port** *port-number*] [**aux**] **interface** *interface-type interface-number*

The OpenFlow function is disabled by default.

1.4.3 Configuring the OpenFlow Multi-controller Mode

1. **Overview**

After the multi-controller mode is configured, multiple controllers can be connected once.

2. **Restrictions and Guidelines**

Before the multi-controller mode is configured, the OpenFlow function must be disabled first. The multi-controller mode can take effect after the OpenFlow function is enabled again.

3. **Procedure**

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Connect an OpenFlow device to the controller.

of mode { **single** | **multiple** }

An OpenFlow device works in multi-controller mode by default.

(4) (Optional) Configure the interval at which an OpenFlow device reconnects to a controller.

of connect-interval *interval*

The default interval at which an OpenFlow device reconnects to a controller is 6 seconds.

(5) (Optional) Configure the echo request sending interval of an OpenFlow device.

of echo-interval *interval*

The default echo request sending interval of an OpenFlow device is 5 seconds.

1.4.4 Configuring the VLAN Tag for Packets Sent to the Controller

1. **Overview**

This function is used to configure a VLAN tag for packets sent to a controller by a device. Packets sent to a controller by a device contain the VLAN tag by default.

2. **Restrictions and Guidelines**

The configuration takes effect immediately.

3. **Procedure**

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a VLAN tag for packets sent by an OpenFlow device to a controller.

of packet vlantag

Packets sent to a controller by an OpenFlow device contain the VLAN tag by default.

1.4.5 Configuring the Table-Lookup Mode

1. **Overview**

This function is used to enable the table-lookup mode for received packets. A device looks up the flow table for the received packets by default, and processes the packets based on the matched entry. If no entry is matched, the packets are discarded.

2. **Restrictions and Guidelines**

The configuration takes effect immediately.

3. **Procedure**

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Connect an OpenFlow device to a controller.

of packet table-lookup { **enable** | **disable** }

The table-lookup mode is enabled for a network device by default.

1.4.6 Configuring the OpenFlow Source IP Address

1. **Overview**

The IP address of a device is configured as a source IP address for connecting to a controller. The default source IP address is the IP address of the connection port.

2. **Restrictions and Guidelines**

The configuration takes effect immediately and the network device is disconnected from the controller. Then, the source IP address is used for reconnection.

3. **Procedure**

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure a source IP address for connecting an OpenFlow device to a controller.

of source-ip *ipv4-address*

The IP address of the connection port is the source IP address for connecting to a controller by default.

1.5 Monitoring

Run the **show** command to check the running status of a configured function to verify the configuration effect.

Table 1-1Monitoring

Command	Purpose
show of	Displays the connection between the OpenFlow device and the controller.
show of port	Displays the port status of the current OpenFlow device.
show of flowtable	Displays the flow table of the current OpenFlow device.
show of group (only available in OpenFlow 1.3)	Displays the group table of the current OpenFlow device.
show of meter (only	Displays the meter table of the current OpenFlow device.

Command	Purpose
available in OpenFlow 1.3)	
show of mergedflow(only available in OpenFlow 1.3)	Displays the merged flow table of the current OpenFlow device.

1.6 Configuration Examples

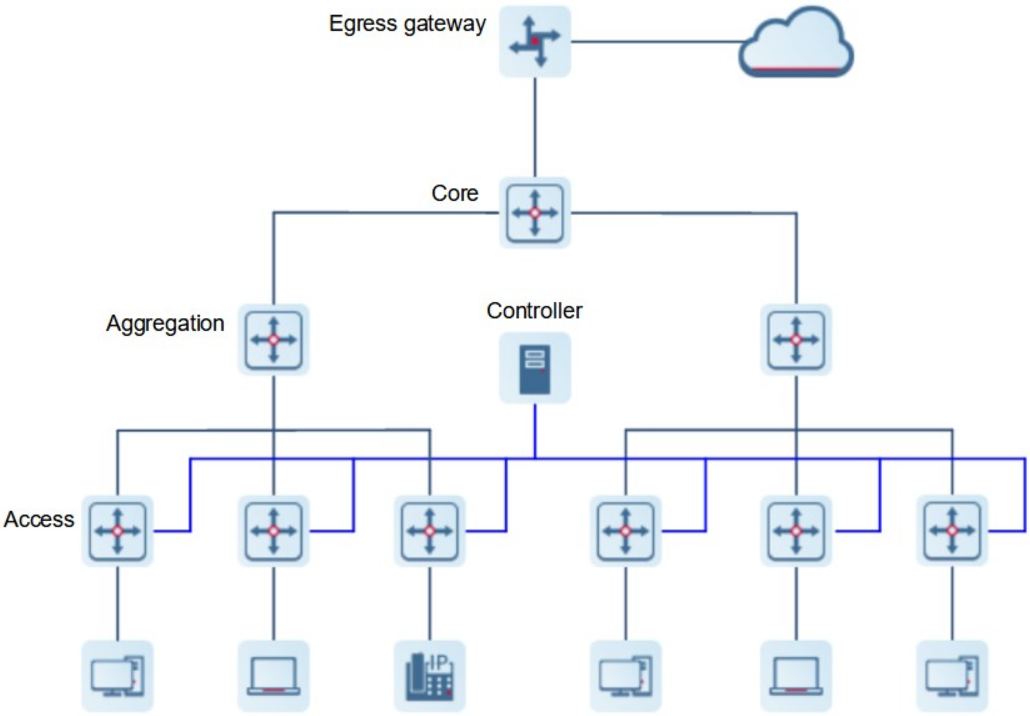
1.6.1 Configuring Central Control

1. **Requirements**

OpenFlow can be used to perform centralized management of access device authentication. As shown in Figure 1-1, a controller is deployed on an access device to authenticate and control the access device, so that the authentication function (on the control plane) is shifted from the access device to the controller.

2. **Topology**

Figure 1-1Topology of Central Control



3. **Notes**

- Enable the OpenFlow function on a device and establish an OpenFlow management channel with a controller.
- Run OpenFlow Client on the access device to interconnect with the controller.
- The controller asks the access device to send an authentication packet to itself over the OpenFlow

protocol.

- The controller completes the authentication process, and sends authentication results to the access device over the OpenFlow protocol to complete admission control on end users.

4. **Procedure**

Enable the OpenFlow function on a device, set the IP address of a controller to 172.18.2.36/24 and the port number to 65535, and disable the LLDP function on the device.

```
Orion_B26Q> enable
Orion_B26Q# configure terminal
Orion_B26Q(config)# no lldp enable
Orion_B26Q(config)# interface mgmt 0
Orion_B26Q(config-if)# ip address 172.18.2.36 255.255.255.0
Orion_B26Q(config-if)# exit
Orion_B26Q(config)# of controller-ip 172.18.2.35 port 65535 interface mgmt 0
```

Configure the single-controller mode on the device.

```
Orion_B26Q(config)# of mode single
```

Remove the VLAN tag from packets sent to the controller.

```
Orion_B26Q(config)# no of packet vlantag
```

Configure the table-lookup mode on the device.

```
Orion_B26Q(config)# ofpacket table-lookup enable
```

Set the OpenFlow source IP address for the device to 192.168.197.25.

```
Orion_B26Q(config)# of source-ip 192.168.197.25
```

5. **Verification**

Display the connection status of OpenFlow 1.3.

```
Orion_B26Q# show of
[0] Controller ID=0 Info=tcp:172.18.2.35 port=6653 interface Mgmt 0, Main is
Connected, Aux is Disabled
```

Display the port status of OpenFlow 1.3.

```
Orion_B26Q# show of port
ID    IFX    INTERFACE            SPEED     LINK    DUPLEX     TX_PKT
RX_PKT       CONFIG
1     1      GigabitEthernet 0/1  Unknown   DOWN    Unknown    0            0
NA
2     2      GigabitEthernet 0/2  Unknown   DOWN    Unknown    0            0
NA
3     3      GigabitEthernet 0/3  Unknown   DOWN    Unknown    0            0
NA
4     4      GigabitEthernet 0/4  Unknown   DOWN    Unknown    0            0
NA
5     5      GigabitEthernet 0/5  Unknown   DOWN    Unknown    0            0
NA
```

```
6     6      GigabitEthernet 0/6  Unknown   DOWN    Unknown    0            0
NA
7     7      GigabitEthernet 0/7  Unknown   DOWN    Unknown    0            0
NA
8     8      GigabitEthernet 0/8  Unknown   DOWN    Unknown    0            0
NA
9     9      GigabitEthernet 0/9  Unknown   DOWN    Unknown    0            0
NA
10    10     GigabitEthernet 0/10 Unknown   DOWN    Unknown    0            0
NA
11    11     GigabitEthernet 0/11 Unknown   DOWN    Unknown    0            0
NA
12    12     GigabitEthernet 0/12 Unknown   DOWN    Unknown    0            0
NA
13    13     GigabitEthernet 0/13 Unknown   DOWN    Unknown    0            0
NA
14    14     GigabitEthernet 0/14 Unknown   DOWN    Unknown    0            0
NA
15    15     GigabitEthernet 0/15 Unknown   DOWN    Unknown    0            0
NA
16    16     GigabitEthernet 0/16 Unknown   DOWN    Unknown    0            0
NA
```

Display the flow table status of OpenFlow 1.3.

```
Orion_B26Q# show of flowtable
/***************** openflow flow table[0]---flow number[2] *******************/

{table="0", duration_sec="5222", priority="0", flags ="0x1",idle_timeout="0",
hard_timeout="0", cookie="0x0", packet_count="0", byte_count="0". match=oxm{all
match} instructions=[apply{acts=[output{port="controller", max_len="3333"}] }] }
xid=472, sync_flag=0x0, ss_index=0x7b6

{table="0", duration_sec="5222", priority="66", flags ="0x1",idle_timeout="0",
hard_timeout="0", cookie="0x0", packet_count="0", byte_count="0".
match=oxm{eth_dst="00:25:64:c4:54:e4", eth_type="0x800", ipv4_dst="1.1.0.0",
ipv4_dst_mask="255.255.240.0"}
instructions=[apply{acts=[output{port="controller", max_len="65535"}] }] }
xid=473, sync_flag=0x0, ss_index=0x7b1

/****************************** openflow flow table end
****************************************/
flow total number = 2
```

Configure the single-controller mode on the device. The device can be connected to only one controller. If more controllers are connected, a notification will be displayed.

```
Orion_B26Q(config)# no of controller-ip
Orion_B26Q(config)# of mode single
```

```
Orion_B26Q(config)# of controller-ip 172.18.122.24 interface gigabitEthernet 0/1
Orion_B26Q(config)# of controller-ip 172.18.122.25 interface gigabitEthernet 0/1
Controller Mode is Single, can't connected
```

Run the **show of** command to check whether the current mode is the table-lookup mode and whether the source IP address is 192.168.197.25.

```
Orion_B26Q# show of
version:openflow1.3, controller[0]:tcp:172.18.105.11 port 6653 interface
GigabitEthernet 1/0/7, main is connected, aux is disable, role is master.
Current controller mode : multiple.
Current packet process mode : Lookup all flow.
Datapath id = 897516188948
Source IP = 192.168.197.25
```

6. **Configuration Files**

```
hostname Device
alias exec sv show version
!
no lldp enable
!
of source-ip 192.168.197.25
of controller-ip 172.18.2.35 port 65535 interface Mgmt 0
no of packet vlantag
!
interface Mgmt 0
 ip address 172.18.2.36 255.255.255.0
 gateway 172.18.2.1
```

7. **Common Errors**

- The controller IP address is incorrectly configured.
- The TCP port of the controller is incorrectly configured.
- The IP address of the local management channel is not configured.