Cisco Plug-in for OpenFlow

Cisco Plug-in for OpenFlow

Cisco Plug-in for OpenFlow, Release 2.0.2 provides better control over networks making them more open, programmable, and application-aware and supports the following specifications defined by the Open Networking Foundation (ONF) standards organization:

- OpenFlow Switch Specification Version 1.0.1 (Wire Protocol 0x01) (referred to as OpenFlow 1.0)
- OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04) (referred to as OpenFlow 1.3).

Prerequisites for Cisco Plug-in for OpenFlow

- A Cisco device and its corresponding operating system that supports the installation of Cisco Plug-in for OpenFlow.

Note A compatibility matrix is delivered with each Cisco application. Refer to this matrix for information about the operating system releases that support features and infrastructure necessary for a particular application, such as Cisco Plug-in for OpenFlow.

- An open virtual application (OVA) package that is compatible with the device operating system and downloaded from an FTP server connected to the device.
- A controller installed on a connected server.

Table 1: Controller Support

OpenFlow Version	Supported Controllers
OpenFlow 1.0	Extensible Network Controller (XNC) 1.0, POX, or Ixia controllers

OpenFlow Version	Supported Controllers
OpenFlow 1.3	Ixia or OpenDaylight

Restrictions for Cisco Plug-in for OpenFlow

- You cannot configure a bridge domain, Virtual LANs, and virtual routing and forwarding (VRF) interfaces on a Cisco Plug-in for OpenFlow logical switch.
- Cisco Plug-in for OpenFlow is not supported on default VDC.
- OpenFlow hybrid switch Integrated model is not supported. OpenFlow hybrid switch (ships-in-the-night) model is supported with physical port separation with virtual device contexts (VDCs). OpenFlow and non-OpenFlow ports must be configured on different VDCs.
- Reachability to controller via Switched Virtual Interface (SVI) is not supported.
- A routing and switching protocol must not be enabled on interfaces that are allocated to OpenFlow VDCs.
- You cannot configure more than 3000 flows in an OpenFlow VDC.

Information About Cisco Plug-in for OpenFlow

Cisco Plug-in for OpenFlow Feature Support

The following is a subset of OpenFlow 1.3 functions that are supported by Cisco Plug-in for OpenFlow.

Supported Feature	Additional Notes
OpenFlow-hybrid switch (ships-in-the-night) type is supported using OpenFlow 1.3 packet format with limitations.	OpenFlow hybrid (ships-in-the-night) hybrid model is supported with physical port seperation on virtual device contexts (VDCs). OpenFlow can be enabled on a subset of devices and ports making a part of the network OpenFlow enabled while the rest of the network continues to run using traditional forwarding principles. But the OpenFlow and non-OpenFlow ports of a device must be configured on different VDCs. OpenFlow hybrid (integrated) switch type is not supported.
Dedicated virtual device context (VDC) for OpenFlow	• OpenFlow can be enabled and installed on up to seven dedicated VDCs if the device has the required space. • A non default VDC must be used for OpenFlow.

Supported Feature	Additional Notes
Connection to up to eight controllers.	• Each Cisco Plug-in for OpenFlow VDC can connect to one controller. You can connect to up to eight controllers using seven VDCS. • Connection is via TCP. • All controllers of a VDC should be running the same OpenFlow version (1.3 or lower).
Pipelines for Cisco Plug-in for OpenFlow logical switch	• Pipelines are mandatory for the logical switch. • The logical switch supports the following pipelines: ◦ Pipeline 321 supports the L2 MAC forwarding table. ◦ Pipeline 322 supports the IPv4 and IPv6 forwarding, ARP, and L2 MAC forwarding tables.
Ethertype selector based table lookup	Ethertype of a packet decides the forwarding table and the corresponding match and action criteria. Ethertype is mandatory for pipeline 322.
Supported Interface Types	Physical interfaces and port-channel interfaces.

Supported Feature	Additional Notes
L2 Forwarding Table (Ethertype = *) (Pipeline 321)	Supported match criteria: • Source MAC address • Destination MAC address • Ethernet type (inner only) • Input port • VLAN priority code point • VLAN ID (with restrictions) **Note** If a packet contains a VLAN tag (Ethertype 0x8100), the outer Ethertype is ignored and the match is done using the VLAN ID, VLAN priority, or Inner Ethertype. Supported action criteria: • Output to multiple ports (supports up to 8 ports) • Output to controller • Set VLAN ID • Strip VLAN ID • Drop
	Supported match criteria: • Ethertype (mandatory) • IP protocol • Layer 4 source port (TCP or UDP) • Layer 4 destination port (TCP or UDP) • Input port Supported action criteria: • Output to multiple ports (supports up to 8 ports) • Punt to controller **Note** Punt to controller cannot be combined with any modify actions. • Set source MAC address (SMAC) • Set destination MAC address (DMAC) • Set VLAN ID • Strip VLAN ID • Drop

Supported Feature	Additional Notes
	Supported match criteria: • Ethertype (mandatory) • IP protocol • Layer 4 source port (TCP or UDP) • Layer 4 destination port (TCP or UDP) • Input port Supported action criteria: • Output to multiple ports (supports up to 8 ports) • Punt to controller **Note** Punt to controller cannot be combined with any modify actions. • Set source MAC address (SMAC) • Set destination MAC address (DMAC) • Set VLAN ID • Strip VLAN ID • Drop
ARP Table (Ethertype = 0x806) (Pipeline 322)	Supported match criteria: • Ethertype (mandatory) • Input port Supported action criteria: • Output to multiple ports (supports up to 8 ports) • Punt to controller • Drop
Default Action	If packets do not match flows of any of the tables above, the default action for each table is as follows: • L2 Forwarding Table-Drop • IPv4 or IPv6 Forwarding Table-Output to port on the same subnet as the destination You can also configure the default action and set it to controller if required.

Supported Feature	Additional Notes
OpenFlow v1.3 message types	The “modify state” and “queue config” message types are not supported. All other message types are supported.
Multiple actions	Flows defined on the controller must follow the guidelines below: • Multiple VLAN actions are not possible. • The flow should not have multiple rewrite actions that override one another the last action is effective. For example, strip VLAN after set VLAN or multiple set VLANs. • You cannot combine an output to port action with a punt to controller or drop action.
OpenFlow 1.3 counters	Per Port—Received Packets, Transmitted Packets, Received Bytes, Transmitted Bytes, Receive Drops, Transmit Drops, Receive Errors, Transmit Errors, Receive Frame Alignment Errors, Receive Overrun Errors, Collisions, Duration (in seconds), Duration (in nanoseconds). **Note** Per Flow and Per Table counters are not supported.

About OpenFlow

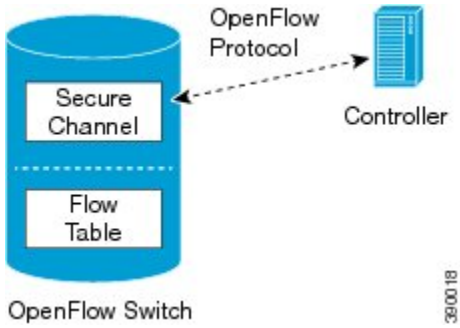
OpenFlow Switch Specification Version 1.0.1 (Wire Protocol 0x01) (referred to as OpenFlow 1.0) and OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04), referred to as OpenFlow 1.3, is based on the concept of an Ethernet switch, with an internal flow table and standardized interface to allow traffic flows on a device to be added or removed. OpenFlow 1.3 defines the communication channel between Cisco Plug-in for OpenFlow and controllers.

Cisco Plug-in for OpenFlow 2.0.2 refers to Cisco Plug-in for OpenFlow, Release 2.0.2.

A controller can be Extensible Network Controller (XNC) 1.0, or any controller compliant with OpenFlow 1.3.

The following figure gives an overview of the OpenFlow network.

Figure 1: OpenFlow Overview



Cisco Plug-in for OpenFlow Operation

Cisco Plug-in for OpenFlow creates OpenFlow–based TCP/IP connections to controllers for a Cisco Plug-in for OpenFlow logical switch. Cisco Plug-in for OpenFlow creates databases for a configured logical switch, OpenFlow-enabled interfaces, and flows. The logical switch database contains all the information needed to connect to a controller. The interface database contains the list of OpenFlow-enabled interfaces associated with a logical switch, and the flow database contains the list of flows on a logical switch as well as for interface that is programmed into forwarded traffic.

OpenFlow Controller Operation

OpenFlow controller (referred to as controller) controls the switch and inserts flows with a subset of OpenFlow 1.3 and 1.0 match and action criteria through Cisco Plug-in for OpenFlow logical switch. Cisco Plug-in for OpenFlow rejects all OpenFlow messages with any other action.

Cisco Plug-in for OpenFlow and Virtual Services Container

Cisco Plug-in for OpenFlow runs in an operating–system–level virtual service container on the device. The Cisco Plug-in for OpenFlow virtual service container is delivered in an open virtual application (OVA) file package (.ova). The OVA package is installed and enabled on the device through the CLI.

How to Configure Cisco Plug-in for OpenFlow

This section includes the following required and optional tasks. All tasks below require the fulfillment of the prerequisites listed in Prerequisites for Cisco Plug-in for OpenFlow, on page 1:

Configuring Physical Device Parameters

This section contains the following:

Configuring Interfaces for a Cisco Plug-in for OpenFlow Logical Switch

You must configure physical interfaces before the interfaces are added as ports of a Cisco Plug-in for OpenFlow logical switch. These interfaces are added as ports of the Cisco Plug-in for OpenFlow logical switch in the Configuring a Cisco Plug-in for OpenFlow Logical Switch , on page 14 section.

Specifying a Route to a Controller

The following tasks are used to specify a route from the device to a controller. This can be done using a physical interface (Front Panel) or a management interface.

- Physical Interface . Refer to Specifying a Route to a Controller Using a Physical Interface, on page 8.
- Management Interface. Refer to Specifying a Route to a Controller Using a Management Interface, on page 10.

The IP address of the controller is configured in the Configuring a Cisco Plug-in for OpenFlow Logical Switch , on page 14 section.

Specifying a Route to a Controller Using a Physical Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **no switchport**
4. **ip address** *ip-address mask*
5. **exit**
6. **ip route 0.0.0.0 0.0.0.0** *next-hop*
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type number* **Example:** `Device(config)# interface Ethernet2/2`	Configures the physical interface. The interface used here should not be a Cisco Plug-in for OpenFlow ports.

	Command or Action	Purpose
Step 3	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures a specified interface as a Layer 3 interface and deletes any interface configuration specific to Layer 2.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.1.4 255.255.255.0`	Configures an IP address for a specified interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**ip route 0.0.0.0 0.0.0.0** *next-hop* **Example:** `Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.1.6`	Configures a default route for packet addresses not listed in the routing table. Packets are directed toward a controller.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the changes persistently by copying the running configuration to the startup configuration.

What to Do Next

Configure interfaces for the Cisco Plug-in for OpenFlow logical switch.

Specifying a Route to a Controller Using a Management Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface mgmt** *management-interface-name number*
3. **ip address** *ip-address mask*
4. **exit**
5. **vrf context management**
6. **ip route 0.0.0.0 0.0.0.0** *next-hop*
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface mgmt** *management-interface-name number* **Example:** `Device(config)# interface mgmt0`	Enters the management interface.
Step 3	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.1.4 255.255.255.0`	Configures an IP address for the interface.
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 5	**vrf context management** **Example:** `Device(config)# vrf context management`	Configures the management Virtual routing and forwarding (VRF) instance and enters in VRF configuration mode.
Step 6	**ip route 0.0.0.0 0.0.0.0** *next-hop* **Example:** `Device(config-vrf)# ip route 0.0.0.0 0.0.0.0 10.0.1.6`	Configures a default route for packet addresses not listed in the routing table. Packets are directed toward a controller.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently by copying the running configuration to the startup configuration.

What to Do Next

Configure interfaces for the Cisco Plug-in for OpenFlow logical switch.

Configuring Interfaces for a Cisco Plug-in for OpenFlow Logical Switch

You must configure physical interfaces before the interfaces are added as ports of a Cisco Plug-in for OpenFlow logical switch. These interfaces are added as ports of the Cisco Plug-in for OpenFlow logical switch in the Configuring a Cisco Plug-in for OpenFlow Logical Switch , on page 14 section.

Configuring a Physical Interface in Layer 2 mode

Perform the following task to add a physical interface to a Cisco Plug-in for OpenFlow logical switch in Layer 2 mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface Ethernet***slot port*
3. **switchport**
4. **switchport mode trunk**
5. **mac packet-classify**
6. **switchport mode trunk allowed vlan** [*vlan-list*]
7. **no shutdown**
8. **end**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface Ethernet***slot port* **Example:** `Device(config)# interface Ethernet2/2`	Specifies the interface for the logical switch and enters interface configuration mode.
Step 3	**switchport** **Example:** `Device(config-if)# switchport`	Specifies an interface as a Layer 2 port.
Step 4	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Specifies an interface as a trunk port. • A trunk port can carry traffic of one or more VLANs on the same physical link. (VLANs are based on the trunk-allowed VLANs list.) By default, a trunk interface carries traffic for all VLANs.
Step 5	**mac packet-classify** **Example:** `Device(config-if)# mac packet-classify`	Enables MAC packet classification on the interface.
Step 6	**switchport mode trunk allowed vlan** [*vlan-list*] **Example:** `Device(config-if)# switchport trunk allowed vlan 1-3`	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 7	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently by copying the running configuration to the startup configuration.

What to Do Next

Repeat these steps to configure any additional interfaces for a Cisco Plug-in for OpenFlow logical switch. Once all the interfaces are configured, install and activate Cisco Plug-in for OpenFlow.

Configuring a Physical Interface in Layer 3 mode

Perform the task below to add a physical interface to a Cisco Plug-in for OpenFlow logical switch in Layer 3 mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **no shutdown**
4. **end**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 2	**interface** *type slot/port*
	
	Example:
	`Device(config)# interface Ethernet1/1`
	`Device(config)# interface port-channel 101`
		Specifies the interface for the logical switch and enters interface configuration mode.

Step 3	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.
Step 4	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently by copying the running configuration to the startup configuration.

What to Do Next

Repeat these steps to configure any additional interfaces for a Cisco Plug-in for OpenFlow logical switch. Once all the interfaces are configured, install and activate Cisco Plug-in for OpenFlow.

Installing and Activating Cisco Plug-in for OpenFlow

Cisco Plug-in for OpenFlow is an application that runs at the operating–system-level virtual services container on a device. Cisco Plug-in for OpenFlow is delivered in an open virtual application (OVA) package. The OVA package is installed and activated on the device through the CLI.

Before installing and activating Cisco Plug-in for OpenFlow, ensure that an OVA package compatible with the device exists on a connected FTP server. Refer to the Prerequisites for a Virtual Services Container. A reload of the device is not essential after installing, uninstalling, or upgrading Cisco Plug-in for OpenFlow software.

To install and activate Cisco Plug-in for OpenFlow software, refer to the instructions in Installing and Activating an Application in a Virtual Services Container, where the virtual services application argument, *virtual-services-name*, can be specified as openflow_plugin.

To uninstall and deactivate Cisco Plug-in for OpenFlow software, refer to the instructions in Deactivating and Uninstalling an Application from a Virtual Services Container, where the virtual services application argument, *virtual-services-name*, must be the same as that specified during installation.

To upgrade Cisco Plug-in for OpenFlow software, refer to the instructions in Upgrading an Application in a Virtual Services Container, where the virtual services application argument, *virtual-services-name*, must be the same as that specified during installation.

Once installed, configure a Cisco Plug-in for OpenFlow logical switch.

Configuring a Cisco Plug-in for OpenFlow Logical Switch

This task configures a Cisco Plug-in for OpenFlow logical switch and the IP address of a controller.

SUMMARY STEPS

1. **configure terminal**
2. **openflow**
3. **switch** *logical-switch-id*
4. **pipeline** *pipeline-id*
5. Do one of the following:
   - **of-port interface** *interface-name*
6. **protocol-version** *version-info*
7. **controller ipv4** *ip-address* [**port** *tcp-port*] [**vrf** *vrf-name*] **security**{**none** | **tls**}
8. (Optional) **logging flow-mod**
9. (Optional) **probe-interval** *probe-interval*
10. (Optional) **rate-limit packet_in** *controller-packet-rate* **burst** *maximum-packets-to-controller*
11. (Optional) **max-backoff** *backoff-timer*
12. **end**
13. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**openflow** **Example:** `Device(config)# openflow`	Enters Cisco Plug-in for OpenFlow mode.
Step 3	**switch** *logical-switch-id* **Example:** `Device(config-ofa)# switch 1`	Specifies an ID for a logical switch that is used for Layer 2 (default) switching operations and enters logical switch configuration mode. • The only logical switch ID supported is 1.
Step 4	**pipeline** *pipeline-id* **Example:**	Configures a pipeline . • This step is mandatory for a logical switch configuration. • You can view the supported pipeline values using the **show openflow hardware capabilities** command. • The valid values are from 321 and 322.

	Command or Action	Purpose
Step 5	Do one of the following: • **of-port interface** *interface-name* **Example:** For a physical interface:	Configures an Ethernet interface interface as a port of a Cisco Plug-in for OpenFlow logical switch. • Do not abbreviate the interface type. Ensure that the interface type is spelled out completely and is as shown in the examples. If the keyword is abbreviated, the interface is not configured. • The interface must be designated for the Cisco Plug-in for OpenFlow logical switch only. • The **mode openflow** configuration is added to an interface when an interface is configured as a port of Cisco Plug-in for OpenFlow. To add or remove an interface as a port of Cisco Plug-in for OpenFlow, ensure that the Cisco Plug-in for OpenFlow is activated and running to ensure the proper automatic addition and removal of the **mode openflow** configuration. To remove an interface as a port of Cisco Plug-in for OpenFlow, use the no form of this command. • Repeat this step to configure additional interfaces.
Step 6	**protocol-version** *version-info* **Example:** `Device(config-openflow-switch)# protocol-version 1.0`	Configures the protocol version. • Supported values are: ◦ **1.0—**Configures device to connect to 1.0 controllers only ◦ **1.3—**Configures device to connect to 1.3 controllers only ◦ **negotiate—**Negotiates the protocol version with the controller. Device uses 1.3 for negotiation. **Note** The default value is **negotiate**. • **drop** is the default action for both tables or pipeline 1. This can be overridden by this configuration or the controller.
Step 7	**controller ipv4** *ip-address* [**port** *tcp-port*] [**vrf** *vrf-name*] **security**{**none** | **tls**} **Example:** Controller in default VRF: `Device(config-openflow-switch)# controller ipv4 10.1.1.2 security none`	Specifies the IPv4 address, port number, and VRF of a controller that can manage the logical switch, port number used by the controller to connect to the logical switch and the VRF of the controller. • If unspecified, the default VRF is used. • Controllers use TCP port 6653 by default. • You can configure up to eight controllers. Repeat this step if you need to configure additional controllers. • If TLS is not disabled in this step, configure TLS trustpoints in the next step. • You can use the **clear openflow switch 1 controller all** command to clear controller connections. This command can reset a connection after

	Command or Action	Purpose
		Transport Layer Security (TLS) certificates and keys are updated. This is not required for TCP connections. A connection to a controller is initiated for the logical switch.
Step 8	**logging flow-mod** **Example:** `Device(config-ofa-switch)# logging flow-mod`	(Optional) Enables logging of flow changes, including addition, deletion, and modification of flows. • Logging of flow changes is disabled by default. • Flow changes are logged in syslog and can be viewed using the **show logging** command. • Logging of flow changes is a CPU intensive activity and should not be enabled for networks greater than 1000 flows.
Step 9	**probe-interval** *probe-interval* **Example:** `Device(config-openflow-switch)# probe-interval 5`	(Optional) Configures the interval, in seconds, at which the controller is probed. • The default value is 5. • The range is from 5 to 65535.
Step 10	**rate-limit packet_in** *controller-packet-rate* **burst** *maximum-packets-to-controller* **Example:** `Device(config-openflow-switch)# rate-limit packet_in 1 burst 4`	(Optional) Configures the maximum packet rate of the connection to the controller and the maximum packets permitted in a burst of packets sent to the controller in a second. • The default value is zero, meaning that an indefinite packet rate and packet burst are permitted. • This rate limit is for Cisco Plug-in for OpenFlow. It is not related to the rate limit of the device (data plane) configured by COPP.
Step 11	**max-backoff** *backoff-timer* **Example:** `Device(config-openflow-switch)# max-backoff 8`	(Optional) Configures the time, in seconds, for which the device must wait before attempting to initiate a connection with the controller. • The default value is eight. • The range is from 1 to 65535.
Step 12	**end** **Example:** `Device(config-openflow-switch)# end`	Exits logical switch configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 13	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently by copying the running configuration to the startup configuration.

What to Do Next

Verify Cisco Plug-in for OpenFlow.

Verifying Cisco Plug-in for OpenFlow

SUMMARY STEPS

1. **show openflow copyright**
2. **show openflow switch** *switch-id*
3. **show openflow switch** *switch-id* **controllers** [**stats**]
4. **show openflow switch** *switch-id* **ports [hidden]**
5. **show openflow switch** *switch-id* **flows** [**table-id** *table-id*][**configured | controller | default | fixed | pending | pending-del**] [**brief | summary**]
6. **show openflow switch** *switch-id* **stats**
7. **show interfaces** *type number* **counters**
8. **show logging last** *number-of-lines*
9. **show running-config | section openflow**
10. **show openflow hardware capabilities**

DETAILED STEPS

Step 1 **show openflow copyright**
Displays copyright information related to Cisco Plug-in for OpenFlow.

Example:
```
Device# show openflow copyright

Cisco Plug-in for OpenFlow
TAC support: http://www.cisco.com/tac
Copyright (c) 2013-2015 by Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0, the GNU
Lesser General Public License (LGPL) Version 2.1, or or the GNU
Library General Public License (LGPL) Version 2. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
```

```
http://www.opensource.org/licenses/lgpl-2.1.php and
http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt
```

Step 2 **show openflow switch** *switch-id*
Displays information related to Cisco Plug-in for OpenFlow logical switch.

Example:

Step 3 **show openflow switch** *switch-id* **controllers** [**stats**]
Displays information related to the connection status between an Cisco Plug-in for OpenFlow logical switch and connected controllers.

Example:

```
Device# show openflow switch 1 controllers

Logical Switch Id: 1
Total Controllers: 3
  Controller: 1
    10.1.1.2:6653
    Protocol: tcp
    VRF: default
    Connected: No
    Role: Master
    Negotiated Protocol Version: disconnected
    Last Alive Ping: N/A
    last_error:No route to host
    state:BACKOFF

  Controller: 2
    5.30.26.111:6800
    Protocol: tcp
    VRF: management
    Connected: No
    Role: Master
    Negotiated Protocol Version: disconnected
    Last Alive Ping: N/A
    last_error:Connection timed out
    state:CONNECTING
    sec_since_disconnect:14

  Controller: 3
    10.1.1.2:6653
    Protocol: tcp
    VRF: management
    Connected: No
    Role: Master
    Negotiated Protocol Version: disconnected
    Last Alive Ping: N/A
    last_error:Connection timed out
    state:CONNECTING
    sec_since_disconnect:13
```

The above sample output is displayed when controller is not yet connected.

```
Device# show openflow switch 1 controllers stats

Logical Switch Id: 1
Total Controllers: 3
  Controller: 1
    address                          :  tcp:10.1.1.2:6653
    connection attempts              :  3009
    successful connection attempts   :  0
    flow adds                        :  0
    flow mods                        :  0
    flow deletes                     :  0
    flow removals                    :  0
    flow errors                      :  0
```

```
    flow unencodable errors         :  0
    total errors                    :  0
    echo requests                   :  rx: 0, tx: 0
    echo reply                      :  rx: 0, tx: 0
    flow stats                      :  rx: 0, tx: 0
    barrier                         :  rx: 0, tx: 0
    packet-in/packet-out            :  rx: 0, tx: 0

  Controller: 2
    address                         :  tcp:5.30.26.111:6800%management
    connection attempts             :  1506
    successful connection attempts  :  0
    flow adds                       :  0
    flow mods                       :  0
    flow deletes                    :  0
    flow removals                   :  0
    flow errors                     :  0
    flow unencodable errors         :  0
    total errors                    :  0
    echo requests                   :  rx: 0, tx: 0
    echo reply                      :  rx: 0, tx: 0
    flow stats                      :  rx: 0, tx: 0
    barrier                         :  rx: 0, tx: 0
    packet-in/packet-out            :  rx: 0, tx: 0

  Controller: 3
    address                         :  tcp:10.1.1.2:6653%management
    connection attempts             :  1506
    successful connection attempts  :  0
    flow adds                       :  0
    flow mods                       :  0
    flow deletes                    :  0
    flow removals                   :  0
    flow errors                     :  0
    flow unencodable errors         :  0
    total errors                    :  0
    echo requests                   :  rx: 0, tx: 0
    echo reply                      :  rx: 0, tx: 0
    flow stats                      :  rx: 0, tx: 0
    barrier                         :  rx: 0, tx: 0
    packet-in/packet-out            :  rx: 0, tx: 0
```

Step 4 **show openflow switch** *switch-id* **ports [hidden]**

Displays the mapping between physical device interfaces and ports of an Cisco Plug-in for OpenFlow logical switch.

Example:

Step 5 **show openflow switch** *switch-id* **flows** [**table-id** *table-id*][**configured | controller | default | fixed | pending | pending-del**] [**brief | summary**]

Displays flows defined for the device by controllers.

Example:

```
Device# show openflow switch 1 flows configured

Logical Switch Id: 1
Total flows: 1

Flow: 1
  Match:
  Actions:           drop
  Priority:          0
  Table:             0
  Cookie:            0x0
  Duration:          1937.586s
  Number of packets: 0
  Number of bytes:   0
```

```
Device# show openflow switch 1 flows fixed

Logical Switch Id: 1
Total flows: 0
```

Step 6 **show openflow switch** *switch-id* **stats**
Displays send and receive statistics for each port defined for a Cisco Plug-in for OpenFlow logical switch.

Example:

Step 7 **show interfaces** *type number* **counters**
Displays send and receive statistics for the specified port defined for an Cisco Plug-in for OpenFlow logical switch.

Example:

Step 8 **show logging last** *number-of-lines*
Displays logging information of flow changes, including addition, deletion or modification of flows.

Step 9 **show running-config | section openflow**
Displays configurations made for Cisco Plug-in for OpenFlow.

Example:

Step 10 **show openflow hardware capabilities**
Displays Cisco Plug-in for OpenFlow configurations.

Example:

Configuration Examples for Cisco Plug-in for OpenFlow

Example: Specifying a Route to a Controller Using a Physical Interface

Example: Installing and Activating Cisco Plug-in for OpenFlow

Refer to *Installing and Activating an Application in a Virtual Services Container* for an example of installing and activating Cisco Plug-in for OpenFlow in a virtual services container of a device.

Additional Information for Cisco Plug-in for OpenFlow

Related Documents

Related Topic	Document Title
Cisco commands	

Standards and RFCs

Standard/RFC	Title
OpenFlow 1.3	*OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04).*
OpenFlow 1.0	*OpenFlow Switch Specification Version 1.0.1 (Wire Protocol 0x01).*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation and tools. Use these resources to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco Plug-in for OpenFlow

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 2: Feature Information for Cisco Plug-in for OpenFlow

Feature Name	Releases	Feature Information
Cisco Plug-in for OpenFlow	Cisco Plug-in for OpenFlow Release 2.0.2	Cisco Plug-in for OpenFlow supports OpenFlow 1.0 and helps networks become more open, programmable, and application-aware.