Cisco OpenFlow Agent for Nexus 3000 and 9000 Series Switches

First Published: 2016-10-30

Last Modified: 2020-09-18

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface includes the following sections:

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Virtual machine installation and administration
- Server administration
- Switch and network administration

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

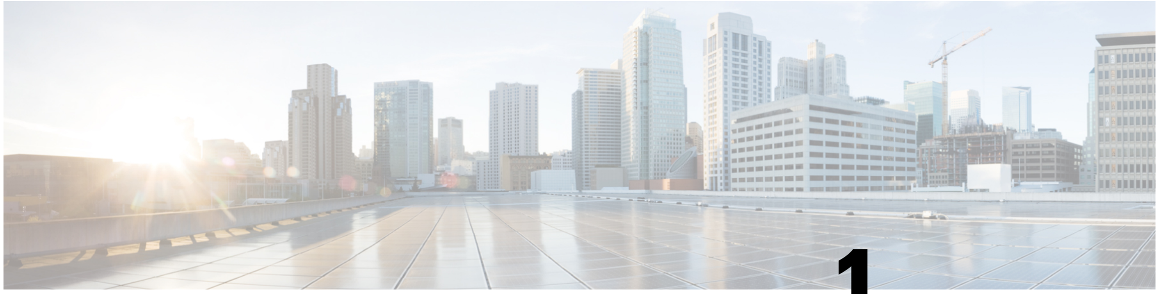
Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexusopenflow-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.



CHAPTER 1

Overview of the Cisco OpenFlow Agent

About Cisco OpenFlow Agent

OpenFlow is an open standardized interface that allows a software-defined networking (SDN) controller to manage the forwarding plane of a network.

Cisco OpenFlow Agent 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

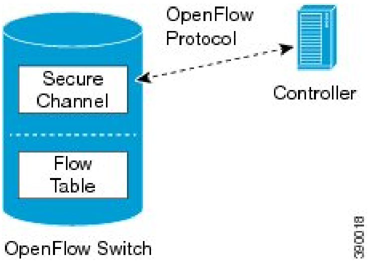
These specifications are based on the concept of an Ethernet switch, with an internal flow table and standardized interface to allow traffic flows on a device to be added or removed. OpenFlow 1.3 defines the communication channel between Cisco OpenFlow Agent and controllers.

A controller can be Cisco Open SDN Controller, or any controller compliant with OpenFlow 1.3.

In an OpenFlow network, Cisco OpenFlow Agent exists on the device and controllers exist on a server that is external to the device. Flow management and any network management are either part of a controller or accomplished through a controller. Flow management includes the addition, modification, or removal of flows, and the handling of OpenFlow error messages.

The following figure gives an overview of the OpenFlow network.

Figure 1: OpenFlow Overview



Cisco OpenFlow Agent Operation

Cisco OpenFlow Agent creates OpenFlow–based TCP/IP connections to controllers for a Cisco OpenFlow Agent logical switch. Cisco OpenFlow Agent 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 OpenFlow Agent logical switch. Cisco OpenFlow Agent rejects all OpenFlow messages with any other action.

OpenFlow Multiple Sub-Switch Operation

For more granular and distributed flow control, you can define multiple virtual subswitches, each with its own controller, its own unique VLAN range, and its own flow control configuration. The controller of a subswitch has configuration access only to the flows of that subswitch. VLANs associated with a subswitch cannot also be associated to another subswitch, and VLAN ranges cannot overlap between subswitches.

When you define one or more subswitches, a lower priority primary switch is implicitly created. A flow is evaluated for a match first on the subswitches and lastly on the primary switch if no previous match was found. There are no default flows (miss-action) for the subswitches.

Information About Cisco OpenFlow Agent

Prerequisites for Cisco OpenFlow Agent

Cisco OpenFlow Agent requires the following conditions:

- A Cisco device that supports Cisco OpenFlow Agent.

  The Supported Platforms for Cisco OpenFlow Agent, on page 53 provides a table of OpenFlow support on Cisco Nexus 9000 and Nexus 3000 Series switches.

- Cisco NX-OS software supports the Cisco OpenFlow Agent.

  The Cisco OpenFlow Agent was introduced in Cisco NX-OS Release 7.0(3)I5(1), replacing the Cisco Plug-in for OpenFlow used in previous releases. The Cisco Plug-in for OpenFlow, which runs as an application in a virtual services container, is no longer supported as of this release. When upgrading from a release earlier than Cisco NX-OS Release 7.0(3)I5(1) to Cisco NX-OS Release 7.0(3)I5(1) or a later release, you must deactivate and uninstall the Cisco Plug-in for OpenFlow application from the virtual services container using the procedure that is described in Uninstalling Cisco Plug-in for OpenFlow, on page 55.

- A Cisco Nexus 3000 platform switch must run in Cisco NX-OS 9000 software mode. On the Cisco Nexus 3000 Series switch, the Cisco NX-OS 9000 mode is activated using the CLI command **system switch-mode n9k**.

- The OpenFlow feature is enabled on the Cisco Nexus switch using the CLI command **feature openflow**.
- A controller is installed on a connected server.

Table 1: Controller Support

OpenFlow Version	Supported Controllers
OpenFlow 1.0	Cisco Open SDN Controller or POX controller.
OpenFlow 1.3	Cisco Open SDN Controller, Ixia, OpenDaylight, or Ryu

Restrictions for Cisco OpenFlow Agent

- Cisco OpenFlow Agent supports only a subset of OpenFlow 1.3 and OpenFlow 1.0 functions. For more information, see Feature Support, on page 4.
- You cannot configure more than one Cisco OpenFlow Agent logical switch. The logical switch ID has a value of 1. However, you can configure up to nine logical subswitches in addition to the primary switch.
- OpenFlow hybrid model (ships-in-the-night) is supported. VLANs configured for Cisco OpenFlow Agent logical switch ports should not overlap with regular device interfaces.
- You cannot configure a bridge domain, Virtual LANs, and virtual routing and forwarding (VRF) interfaces on an Cisco OpenFlow Agent logical switch. You can configure only Layer 2 physical interfaces or port-channel interfaces.
- For Cisco Nexus 3000 Series switches, the total number of VLANs across all ports cannot exceed 32000. For example, if you have configured 512 VLANs per port, you cannot configure more than 62 ports (32000/512). If you have configured 4000 VLANs per port, you cannot configure more than 8 ports (32000/4000).
- You cannot configure more than 512 VLANs in Per-VLAN Spanning Tree+ (PVST+) mode.
- The Cisco OpenFlow Agent supports IPv4 and IPv6 flow matching, but not both simultaneously. The choice is configured in the TCAM configuration commands. IPv4 and IPv6 dual stack is not supported.
- For IPv6 OpenFlow, you must explicitly carve the OpenFlow–IPv6 TCAM region.
- ISSU from the previously supported Cisco Plug-in for OpenFlow to the Cisco OpenFlow Agent is not supported.
- MIBs and XMLs are not supported
- Reachability to controller via Switched Virtual Interface (SVI) is not supported.
- The minimum idle timeout for flows must be (2 * statistics collection interval) + 1 second.
- LACP port-channels are not supported for OpenFlow. Remove all OpenFlow related configurations before downgrading to an earlier release.

Feature Support

The following is a subset of OpenFlow 1.3 and OpenFlow 1.0 functions that are supported by Cisco OpenFlow Agent.

Supported Feature	Additional Notes
The OpenFlow hybrid (ships-in-night) model is supported using the OpenFlow packet format	OpenFlow-hybrid models where traffic can flow between Cisco OpenFlow Agent ports and regular interfaces (integrated) are not supported. Both types of ports can transmit and receive packets. **Note** VLANs must be configured such that the VLANs on the Cisco OpenFlow Agent do not overlap with those on the regular device interfaces.
Configuration of port-channel and physical interfaces as Cisco OpenFlow Agent logical switch ports	• Bridge domain, Virtual LANs and Virtual Routing and Forwarding (VRF) interfaces are not supported. • Only L2 interfaces can be Cisco OpenFlow Agent Logical switch ports.
Configuration of VLANs for each port of the Cisco OpenFlow Agent logical switch	Total number of VLANs across all ports cannot exceed 512. Maximum VLAN range supported is 4000. You can configure 8 such ports on the Cisco OpenFlow Agent device. Recommended VLAN range supported is 512. You can configure 62 such ports on the Cisco OpenFlow Agent device. VLAN range greater than 512 is not supported in Per-VLAN Spanning Tree+ (PVST+) mode.
Pipelines for Cisco OpenFlow Agent Logical Switch	• Pipelines are mandatory for the logical switch. • The logical switch supports the following pipelines: • Pipeline 201 supports the L3 ACL forwarding table. • Pipeline 202 supports an L3 ACL forwarding table and an L2 MAC forwarding table. Mandatory matches and actions in both tables must be specified in all configured flows. • Pipeline 205 supports MAC and MAC-IP route tables.

Supported Feature	Additional Notes
L3 ACL Forwarding Table (Match Criteria)	The following match criteria are supported: • Ethertype • Ethernet MAC destination (Double-wide TCAM required) • Ethernet MAC source (Double-wide TCAM required) • VLAN ID (for IPv4 packets only) • VLAN priority (Supported for the Ethertype value 0x0800 (IP) only) • IPv4 source address (Supported for the Ethertype value 0x0800 (IP) only) • IPv4 destination address (Supported for the Ethertype value 0x0800 (IP) only) • IPv6 source address (Supported for the Ethertype value 0x86DD (IP) only) • IPv6 destination address (Supported for the Ethertype value 0x86DD (IP) only) • IP DSCP (Supported for the Ethertype values 0x0800 or 0x86DD (IP) only) • IP protocol (Supported for the Ethertype values 0x0800 or 0x86DD (IP) only) • Layer 4 source port (Supported for the Ethertype values 0x0800 or 0x86DD (IP) only) • Layer 4 destination port (Supported for the Ethertype values 0x0800 or 0x86DD (IP) only)

Supported Feature	Additional Notes
L3 ACL Forwarding Table (Action Criteria)	The following action criteria are supported: • Output to multiple ports • Output to a specified interface • Output to controller (OpenFlow Packet-In message) • Rewrite source MAC address (SMAC) • Supported for the Ethertype values 0x0800 or 0x86DD (IP) only • Rewrite destination MAC address (DMAC) • Supported for the Ethertype values 0x0800 or 0x86DD (IP) only • Rewrite VLAN ID • Supported for the Ethertype values 0x0800 or 0x86DD (IP) only • Strip VLAN (Supported for the Ethertype values 0x0800 or 0x86DD (IP) only) • Drop **Note** Rewrite DMAC and Rewrite SMAC actions must be specified together.
L2 MAC Forwarding Table	Match Criteria: • Destination MAC address (mandatory) • VLAN ID (mandatory) Action Criteria: • Output to multiple ports • Drop
Default Forwarding Rule	All packets that cannot be matched to flows are dropped by default. You can configure sending unmatched packets to the controller.
OpenFlow 1.3 message types	The “modify state” and “queue config” message types are not supported. All other message types are supported.
Connection to up to eight controllers	Transport Layer Security (TLS) is supported for the connection to the controller.

Supported Feature	Additional Notes
Multiple actions	If multiple actions are associated with a flow, they are processed in the order specified. The output action should be the last action in the action list. Any action after the output action is not supported, and can cause the flow to fail and return an error to the controller. Flows defined on the controller must follow the following guidelines : • The flow can have only up to 16 output actions. • The flow should have the output action at the end of all actions. • The flow should not have multiple rewrite actions that override one another. For example, strip VLAN after set VLAN or multiple set VLANs. • The flow should not have an output–to–controller action in combination with other output–to–port actions or with VLAN–rewrite actions. • Flows with unsupported actions will be rejected.
Supported counters	Per Table—Active Entries, Packet Lookups, Packet Matches. Per Flow—Received Packets. Per Port—Received or Transmitted packets, bytes, drops and errors.

CHAPTER 2

Configuring the Cisco OpenFlow Agent

All tasks in this section require the fulfillment of the prerequisites listed in Prerequisites for Cisco OpenFlow Agent, on page 2.

Enabling the Cisco OpenFlow Agent

Enabling the Cisco OpenFlow Agent on the Cisco Nexus 3000 Series Switch

To run the Cisco OpenFlow Agent, a Cisco Nexus 3000 Series switch must run in Cisco NX-OS 9000 software mode. This procedure activates the Cisco Nexus 9000 mode and enables the Cisco OpenFlow Agent.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**system switch-mode n9k** **Example:** `Device(config)# system switch-mode n9k`	Activates the Cisco NX-OS 9000 mode on the Cisco Nexus 3000 Series switch.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	**write erase** **Example:** `Device# write erase`	Erases the startup configuration file. **Note** It is highly recommended to make a backup copy of the running configuration before entering the **write erase** command.
Step 6	**reload** **Example:** `Device# reload`	Reloads the operating system of the device.
Step 7	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode (after reload).
Step 8	**feature openflow** **Example:** `Device(config)# feature openflow`	Enables the Cisco OpenFlow Agent.

What to do next

Adjust the number of flow entries.

Enabling the Cisco OpenFlow Agent on the Cisco Nexus 9000 Series Switch

This procedure enables the Cisco OpenFlow Agent.

Procedure

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

	Command or Action	Purpose
Step 2	**feature openflow** **Example:** `Device(config)# feature openflow`	Enables the Cisco OpenFlow Agent.

What to do next

Adjust the number of flow entries.

Configuring Physical Device Parameters for Cisco Nexus 3000 and 9000 Series Switches

Adjusting the Number of Flow Entries

You can use this task to adjust the number of L3 flow entries.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	Required: **hardware access-list tcam region racl** *size* **Example:** `Device(config)# hardware access-list tcam region racl 0`	Configures the size of TCAM region for router ACLs.
Step 3	Required: **hardware access-list tcam region e-racl** *size* **Example:** `Device(config)# hardware access-list tcam region e-racl 0`	Configures the size of TCAM region for egress router ACLs.
Step 4	Required: **hardware access-list tcam region l3qos** *size* **Example:** `Device(config)# hardware access-list tcam region l3qos 0`	Configures the size of TCAM region for QoS.

	Command or Action	Purpose
Step 5	Required: **hardware access-list tcam region span** *size* **Example:** `Device(config)# hardware access-list tcam region span 0`	Configures the size of TCAM region for SPAN.
Step 6	Required: **hardware access-list tcam region redirect** *size* **Example:** `Device(config)# hardware access-list tcam region redirect 0`	Configures the size of TCAM region for redirects.
Step 7	Required: **hardware access-list tcam region vpc-convergence** *size* **Example:** `Device(config)# hardware access-list tcam region vpc-convergence 0`	Configures the size of TCAM region for virtual port channel (vPC) convergence.

Step 8	Required: Enter one of the following commands:
	• **hardware access-list tcam region openflow** *size* [**double-wide**]
	• **hardware access-list tcam region openflow-ipv6** *size* [**double-wide**]
	Example:
	`Device(config)# hardware access-list tcam region openflow 1024`
	Example:
	`Device(config)# hardware access-list tcam region openflow-ipv6 1024 double-wide`
		Configures the size of TCAM region for interface ACLs. For a TCAM region larger than 256, configure the size in multiples of 512.
		To accommodate the additional match criteria of source and destination MAC addresses, the Cisco Nexus 3000 and 9000 Series switches support a new TCAM region, **double-wide**, which is a double-wide interface ACL. The maximum TCAM size is 3072 for single-wide and 1536 for double-wide.
		For more information, see the following tables for matches and actions supported for Cisco Nexus 9000 Series switches.
		The **openflow-ipv6** option forces the use of the IPv6 stack for OpenFlow.
		Note To activate the TCAM regions, a reload is needed.
		You can view the supported pipeline values by entering the **show openflow hardware capabilities** command.

	Command or Action	Purpose

Table 2: Matches Supported in Cisco Nexus 9000 Series Switches

Packet Match Fields	L3 Table 201	L3 Table 202	L2 Table 202
Source MAC address	✓ (double wide)	✓ (double wide)	
Destination MAC address	✓ (double wide)		✓
Ether type	✓	✓	
VLAN ID	✓	✓	✓
VLAN CoS	✓	✓	
Source IPv4 Address	✓	✓	
Destination IPv4 Address	✓	✓	
Source IPv4 UDP/TCP Port	✓	✓	
Destination IPv4 UDP/TCP Port	✓	✓	
IPv4 DSCP	✓		
Protocol IP	✓		
Input Interface	✓		

Table 3: Action Supported in Cisco Nexus 9000 Series Switches

Actions	L3 Table 201	L3 Table 201	L2 Table 202
Output Interfaces	✓	✓	✓
Punt to Controller	✓	✓	✓
Copy to Controller	✓	✓	
Push VLAN	✓	✓	
POP VLAN	✓	✓	
DROP	✓	✓	✓
Normal Forwarding	✓	✓	✓

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 11	**reload** **Example:** `Device# reload`	Reloads the operating system of a device.

What to do next

Configure global variables for Cisco OpenFlow Agent logical switch.

Configuring Global Variables for Cisco OpenFlow Agent Logical Switch

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	(Optional) **spanning-tree mode mst** **Example:** `Device(config)# spanning-tree mode mst`	Sets the Spanning Tree Protocol (STP) mode to MST. This step is required if you need VLANs more than 512.

Step 3	(Optional) **vlan** {*vlan-id* | *vlan-range*}
	Example:
	`Device(config)# vlan 1-512`
		Adds a VLAN or VLAN range for interfaces on the device and enters the VLAN configuration mode. This step is needed only if VLAN tagging is required.
		• Total number of VLANs across all interfaces cannot exceed 32000.
		• Maximum VLAN range supported is 4000 (in Multiple Spanning Tree [MST] mode).
		• Recommended VLAN range is 512.

Step 4	**exit** **Example:** `Device(config)# exit`	Ends global 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 through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Configure control plane policing for packets sent to a controller.

Cisco OpenFlow Agent for Cisco Nexus 3500 Platform Switches

Guidelines and Limitations for Cisco Nexus 3500 Platform Switches

The following are guidelines and limitations for Cisco Nexus 3500 platform switches:

- Packets incoming with the following etherTypes are treated differently for Cisco Nexus 3500 platform switches as part of OpenFlow. Packets with these etherTypes cannot be matched and forwarded using

OpenFlow rules with match on specific etherTypes, instead the MATCH_ANY rule works under certain conditions as mentioned in the following table. The difference in behavior for processing such packets is mostly due to a limitation with the ASIC.

- 0x22e9
- 0x8035
- 0x8100
- 0x8927
- 0x8926
- 0x8903
- 0x88a8
- 0xfee1
- 0x8808

Table 4: Specific EtherTypes and Behaviors on Cisco Nexus 3500 Platform Switches

SL#	EtherTypes	Purpose	Match Specific EtherType	Remarks
1	0x22e9	CNTag	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works
2	0x8035	RARP	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works

SL#	EtherTypes	Purpose	Match Specific EtherType	Remarks

3	0x8100	Dot1q	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works.
				Special Case:- If VLAN_ID is '0'
				Dot1q header (4 bytes) is removed and packet is forwarded
				`Ingress_pkt [DA+SA+8100+0000+PAYLOAD]`
				`-> switch_3500 -> egress_pkt [DA+SA+PAYLOAD]`
				The VLAN ID 0 is used to send priority-tagged frames. In general, ASIC pipeline this VLAN ID tag to be ignored and the Ethernet frame to be processed according to the priority configured in the 802.1P bits of the 802.1Q Ethernet frame header.

4	0x8808	PauseFrames (FlowControl)	Matches specific Ethertype. Limitation is, stats will not get updated.	Match ANY works Limitation - Stats will not get updated

5	0x8927	CopperLan	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works with the following caveat. 0x8927 header (8 bytes) is removed and the packet is forwarded matching Match-ANY rule.
				`Ingress_pkt [DA+SA+8927+6bytes+PAYLOAD]`
				`→ switch_3500 → egress_pkt [DA+SA+PAYLOAD]`

6	0x8926	Cisco VNTag	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works with the following caveat. VNTag header (6 bytes) is removed and the packet is forwarded matching Match-ANY rule.
				`Ingress_pkt [DA+SA+8926+4bytes+PAYLOAD]`
				`→ switch_3500 → egress_pkt [DA+SA+PAYLOAD]`

7	0x8903	Cisco FabricPath	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works with the following caveat. Outer DCE header (16 bytes) is removed and inner packet gets forwarded matching Match-ANY rule.
				`Ingress_pkt [ODA+OSA+8903+2bytes+IDA+ISA+PAYLOAD]`
				`→ switch_3500 → egress_pkt [IDA+ISA+PAYLOAD]`

SL#	EtherTypes	Purpose	Match Specific EtherType	Remarks
8	0x88a8	QinQ	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works with the following caveat. 0x88a8 etherType is modified to dot1q (0x8100) etherType and forwarded matching Match-ANY rule. `Ingress_pkt → [DA+SA+88a8+TAG+PAYLOAD] → switch_3500 → egress_pkt [DA+SA+8100+TAG+PAYLOAD]`
9	0xfee1	UNKNOWN	Does not match on specific EtherType and default rule to drop gets applied	Match ANY works with the following caveat. 0xfee1 header (8 bytes) is removed and packet is forwarded matching Match-ANY rule. `Ingress_pkt → [DA+SA+fee1+6bytes+DATA] → switch_3500 → egress_pkt [DA+SA+DATA]`
10	0x8903	Encapsulation header with EtherType 0x8903	Does not match 0x8903 EtherType if it is in an encapsulated header as the header is removed.	There is an ASIC limitation for DCE packets with multicast DA being handled in a different way. Packets are flooded out of all active ports instead of being forwarded to specific port as per the OpenFlow flows installed on the switch.

Enabling the Cisco OpenFlow Agent on Cisco Nexus 3500 Platform Switches

This procedure enables the Cisco OpenFlow Agent.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature openflow** **Example:** `switch(config)# feature openflow`	Enables the Cisco OpenFlow Agent.

What to do next

Adjust the number of flow entries.

Enabling Hardware Support for OpenFlow on Cisco Nexus 3500 Platform Switches

Procedure

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

Step 2	Enter one of the following commands:
	• **hardware profile forwarding-mode openflow-hybrid**
	• **hardware profile forwarding-mode openflow-only**
	Example:
	`switch(config)# hardware profile forwarding-mode openflow-hybrid`
	Example:
	`switch(config)# hardware profile forwarding-mode openflow-only`
		The **hardware profile forwarding-mode openflow-hybrid** command sets the OpenFlow hybrid forwarding mode.
		Note In the OpenFlow hybrid model, normal ports and OpenFlow enabled ports can coexist. When using the OpenFlow hybrid model, VLANs configured for OpenFlow logical switch ports must not overlap with normal device interfaces.
		The **hardware profile forwarding-mode openflow-only** command set the OpenFlow only forwarding mode.
		Note In this mode, all available ports are considered a part of OpenFlow-based forwarding.

Step 3	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 4	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	**reload** **Example:** `switch# reload`	Reloads the operating system of a device.

Enabling Re-Direct Control Plane Packets for OpenFlow Ports on the Cisco Nexus 3500

The **hardware profile openflow forward-pdu** command is introduced in the Cisco NX-OS 9.3(5) release to forward link-level PDUs. With this CLI, the behavior of PDUs with destinations the same as one of the following MAC addresses skip punt-to-CPU and honor configured OpenFlow rules. There is no change in the behavior of other Layer 2 or Layer 3 protocol packets.

```
0180.c200.0000
0180.c200.0002
0100.0ccc.cccc
0100.0ccc.cccd
```

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**hardware profile openflow forward-pdu** **Example:** `switch(config)# hardware profile openflow forward-pdu`	Configures the protocol data unit.
Step 3	(Optional) **no hardware profile openflow forward-pdu** **Example:** `switch(config)# no hardware profile openflow forward-pdu`	Removes the protocol data unit configuration.

Configuring Global Variable for Cisco OpenFlow Agent Logical Switch for Cisco Nexus 3500

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	(Optional) **spanning-tree mode mst** **Example:** `Device(config)# spanning-tree mode mst`	Sets the Spanning Tree Protocol (STP) mode to MST. This step is required if you need VLANs more than 512.

	Command or Action	Purpose

Step 3	(Optional) **vlan** {*vlan-id* | *vlan-range*}
	Example:
	`Device(config)# vlan 1-512`
		Adds a VLAN or VLAN range for interfaces on the device and enters the VLAN configuration mode. This step is needed only if VLAN tagging is required.
		• Total number of VLANs across all interfaces cannot exceed 32000.
		• Maximum VLAN range supported is 4000 (in Multiple Spanning Tree [MST] mode).
		• Recommended VLAN range is 512.

Step 4	**exit** **Example:** `Device(config)# exit`	Ends global 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 through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Configure control plane policing for packets sent to a controller.

Configuration Examples for the Cisco Nexus 3500 Platform Switches

Example Enabling Cisco OpenFlow Agent in the Cisco Nexus 3500 Platform Switches

```
Device# configure terminal
Device(config)# feature openflow
Device(config)# show feature | inc openflow
openflow               1          enabled

Enter either of the following commands at the prompt to configure OpenFlow TCAM:

Device(config)# hardware profile forwarding-mode openflow-hybrid

Device(config)# hardware profile forwarding-mode openflow-only

Device(config)# exit
Device# copy running-config startup-config
Device# reload
```

Example: Cisco OpenFlow Agent Logical Switch Configuration (Default VRF) for Cisco Nexus 3500 Platform Switches

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 203
! Specifies the pipeline that enables the IP Forwarding Table.
Device(config-ofa-switch)# logging flow-mod
Device(config-ofa-switch)# max-backoff 5
Device(config-ofa-switch)# probe-interval 5
Device(config-ofa-switch)# rate-limit packet-in 300 burst 50
Device(config-ofa-switch)# controller ipv4 10.0.1.6 security none
! Adding an interface to the Cisco OpenFlow Agent logical switch.
Device(config-ofa-switch)# of-port interface ethernet1/1
Device(config-ofa-switch)# of-port interface ethernet1/2
! Adding a port channel to the Cisco OpenFlow Agent switch.
Device(config-ofa-switch)# of-port interface port-channel 2
Device(config-ofa-switch)# end
Device# copy running-config startup-config
```

Example: Configuring a Cisco OpenFlow Agent Logical Switch (Management VRF) for Cisco Nexus 3500 Platform Switches

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 203
! Specifying a controller that is part of a VRF.
Device(config-ofa-switch)# controller ipv4 10.0.1.6 vrf management security none
! Adding an interface to the Cisco OpenFlow Agent logical switch.

Device(config-ofa-switch)# of-port interface ethernet1/1
Device(config-ofa-switch)# of-port interface ethernet1/2
! Adding a port channel to the Cisco OpenFlow Agent switch.
Device(config-ofa-switch)# of-port interface port-channel 2
Device(config-ofa-switch)# end
Device# copy running-config startup-config
```

Example: Creating a Sub-Switch for Cisco Nexus 3500 Platform Switches

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 203
Device(config-ofa-switch)# controller ipv4 5.30.199.200 port 6645 vrf management security
none
Device(config-ofa-switch)# of-port interface port-channel1000
Device(config-ofa-switch)# of-port interface Ethernet1/1
Device(config-ofa-switch)# of-port interface Ethernet1/37
Device(config-ofa-switch)# of-port interface Ethernet1/39
Device(config-ofa-switch)# logging flow-mod
Device(config-ofa-switch)# sub-switch 2 vlan 100
Device(config-ofa-switch-subswitch)# controller ipv4 5.30.19.239 port 6653 vrf management
security none
```

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 23.

- Management Interface. Refer to Specifying a Route to a Controller Using a Management Interface, on page 24.

The IP address of the controller is configured in the Configuring a Cisco OpenFlow Agent Logical Switch , on page 27 section.

Specifying a Route to a Controller Using a Physical Interface

Procedure

	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 Ethernet1/1`	Enters the physical interface. The interface used here should not be a Cisco OpenFlow Agent port.
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	**ping** *controller-ip-address* **Example:** `Device(config)# ping 192.0.20.123`	Ping your controller to verify a working route.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Specify a route to a controller using a management interface.

Specifying a Route to a Controller Using a Management Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *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.
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.

	Command or Action	Purpose
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Configure interfaces for the Cisco OpenFlow Agent logical switch.

Configuring Interfaces for a Cisco OpenFlow Agent Logical Switch

You must configure physical or port-channel interfaces before the interfaces are added as ports of a Cisco OpenFlow Agent logical switch. These interfaces are added as ports of the Cisco OpenFlow Agent logical switch in the Configuring a Cisco OpenFlow Agent Logical Switch , on page 27 section.

Configuring a Physical Interface in Layer 2 mode

Perform the task below to add a physical interface to a Cisco OpenFlow Agent logical switch in Layer 2 mode.

Procedure

	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 Ethernet1/23`	Specifies the interface for the logical switch and enters interface configuration mode.
Step 3	(Optional) **channel-group** *group-number* **Example:** `Device(config-if)# channel-group 2`	Adds the interface to a port-channel.
Step 4	Required: **switchport** **Example:** `Device(config-if)# switchport`	Specifies an interface as a Layer 2 port.
Step 5	Required: **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

	Command or Action	Purpose
		VLANs list.) By default, a trunk interface carries traffic for all VLANs.
Step 6	Required: **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 through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Repeat these steps to configure any additional interfaces for a Cisco OpenFlow Agent logical switch.

Configuring a Port-Channel Interface

Perform the task below to create a port-channel interface for a Cisco OpenFlow Agent logical switch.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface port-channel** *number* **Example:** `Device(config)# interface port-channel 2`	Specifies the interface for the logical switch and enters interface configuration mode.
Step 3	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Specifies the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are

	Command or Action	Purpose
		based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. **Note** If the port-channel is specified as a trunk interface, ensure that member interfaces are also configured as trunk interfaces.
Step 4	Required: **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 5	**end**
	Example:
	`Device(config-if)#` **`end`**
		Ends interface configuration mode and enters privileged EXEC mode.

Step 6	**copy running-config startup-config** **Example:** `Device#` **`copy running-config startup-config`**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Activate Cisco OpenFlow Agent.

Configuring a Cisco OpenFlow Agent Logical Switch

This task configures a Cisco OpenFlow Agent logical switch and the IP address of a controller.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**openflow** **Example:** `Device(config)#` **`openflow`**	Enters OpenFlow configuration mode.
Step 3	Required: **switch** *switch-id* **pipeline** *pipeline-id*	Creates an OpenFlow switch with a pipeline.

	Command or Action	Purpose
	Example: `Device(config-ofa)# **switch 1 pipeline 201**`	- This step is mandatory for a logical switch configuration. - You can view the supported pipeline values using the **show openflow hardware capabilities** command. **Note** For the Cisco Nexus 3500 platform switches, the value of *pipeline-id* is 203.

Step 4	Enter one of the following commands:
	- **of-port interface** *interface-name*
	- **of-port interface** *port-channel-name*
	Example:
	For a physical interface:
	`Device(config-ofa-switch)# **of-port interface ethernet1/1**`
	For a port-channel interface:
	`Device(config-ofa-switch)# **of-port interface port-channel2**`
		Configures an Ethernet interface or port-channel interface as a port of a Cisco OpenFlow Agent logical switch.
		- Standard Cisco NX-OS interface type abbreviations are supported.
		- The interface must be designated for the Cisco OpenFlow Agent logical switch only.
		- The **mode openflow** configuration is added to an interface when an interface is configured as a port of Cisco OpenFlow Agent. To add or remove an interface as a port of Cisco OpenFlow Agent, ensure that the Cisco OpenFlow Agent 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 OpenFlow Agent, use the **no** form of this command.
		- An interface configured for a port channel should not be configured as a Cisco OpenFlow Agent logical switch port.
		- Repeat this step to configure additional interfaces.

Step 5	**controller ipv4** *ip-address* [**port** *tcp-port*] [**vrf** *vrf-name*] **security**{**none** | **tls**}
	Example:
	Controller in default VRF:
	`Device(config-ofa-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.

	Command or Action	Purpose
		• 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 using the **tls** command. • You can use the **clear openflow switch 1 controller all** command to clear controller connections. This command can reset a connection after 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 6	(Optional) **tls trust-point local** *local-trust-point* **remote** *remote-trust-point* **Example:** `Device(config-ofa-switch)# **tls trust-point local mylocal remote myremote**`	Specifies the local and remote TLS trustpoints to be used for the controller connection. • For information on configuring trustpoints, refer to the "Configuring PKI" chapter of the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*.
Step 7	(Optional) **logging flow-mod** **Example:** `Device(config-ofa-switch)# **logging flow-mod**`	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 8	(Optional) **probe-interval** *probe-interval* **Example:** `Device(config-ofa-switch)# **probe-interval 5**`	Configures the interval, in seconds, at which the controller is probed with echo requests. • The default value is 5. • The range is from 5 to 65535.
Step 9	(Optional) **rate-limit packet_in** *controller-packet-rate* **burst** *maximum-packets-to-controller*	Configures the maximum packet rate of the connection to the controller and the maximum

	Command or Action	Purpose
	Example: `Device(config-ofa-switch)# rate-limit packet_in 300 burst 50`	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 OpenFlow Agent. It is not related to the rate limit of the device (data plane) configured by COPP.
Step 10	(Optional) **max-backoff** *backoff-timer* **Example:** `Device(config-ofa-switch)# max-backoff 8`	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 11	(Optional) **datapath-id** *id* **Example:** `Device(config-ofa-switch)# datapath-id 0x111`	*id* is a 64bit hex value. A valid *id* is in the range [0x1-0xffffffffffffffff]. This identifier allows the controller to uniquely identify the device.
Step 12	(Optional) **protocol-version** [**1.0** | **1.3** | **negotiate**] **Example:** `Device(config-ofa-switch)# protocol-version 1.3`	This command forces a specific version of the controller connection. If you force version 1.3 and the controller supports only 1.0, no session is established (or vice versa). The default behavior is to negotiate a compatible version between the controller and device. 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**—(Default) Negotiates the protocol version with the controller. The device uses version 1.3 for negotiation.
Step 13	(Optional) **shutdown** **Example:** `Device(config-ofa-switch)# shutdown`	This disables the OpenFlow switch without having to remove all the other configuration.
Step 14	Required: **default-miss** *value* **Example:**	The **default-miss** command sets the behavior when a packet does not match a flow in the flow table. The controller flows may override default-miss flows.

	Command or Action	Purpose
	`Device(config-ofa-switch)#` **`default-miss continue-normal`**	**Note** Not every action is supported on every platform. **continue-drop**: a miss in a flow table will cascade to perform a match in the next table (if applicable). A miss in the terminal table in the pipeline will result in the packet being dropped. **continue-normal**: a miss in a flow table will cascade to perform a match in the next table (if applicable). A miss in the terminal table in the pipeline will result in the packet being sent to the switch's normal hardware processing. **continue-controller**: a miss in a flow table will cascade to perform a match in the next table (if applicable). A miss in the terminal table in the pipeline will result in the packet being sent to the controller. **drop**: a miss in the first flow table of the pipeline will not cascade to any other table. Instead the packet will be dropped. **normal**: a miss in the first flow table of the pipeline will not cascade to any other table. Instead the packet will be sent to the switch's normal hardware forwarding. **controller**: a miss in the first flow table of the pipeline will not cascade to any other table. Instead the packet will be sent to the controller.
Step 15	(Optional) **statistics collection-interval** *seconds* **Example:** `Device(config-ofa-switch)#` **`statistics collection 10`**	A setting of zero disables statistics collection. If collection is enabled, the interval must be a minimum of seven seconds. The interval setting can be used to reduce the CPU load from periodic statistics polling. For example, if you have 1000 flows and choose a statistics collection interval of 10 seconds, 1000flows/10s = 100 flows per second poll rate.

	Command or Action	Purpose
		Note Each flow table has a prescribed maximum flows-per-second poll rate supported by hardware as displayed in the **show openflow hardware capabilities** command. If you choose a statistics collection interval that is too small, the maximum rate supported by the hardware is used, effectively throttling the statistics collection.

Step 16	**end**
	Example:
	`Device(config-ofa-switch)# end`
		Exits logical switch configuration mode and enters privileged EXEC mode.

Step 17	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to do next

Configure logical sub-switches.

Configuring Logical Sub-Switches

This task configures a logical subswitch for OpenFlow control by a controller other than the primary controller.

Before you begin

Configure an OpenFlow logical switch.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**openflow** **Example:** `Device(config)# openflow`	Enters OpenFlow configuration mode.

	Command or Action	Purpose
Step 3	Required: **switch** *switch-id* **pipeline** *pipeline-id* **Example:** `Device(config-ofa)#` **`switch 1 pipeline 201`**	Selects the existing OpenFlow switch under which the subswitch will be created. This is the primary switch, which has the ID of 1. **Note** For the Cisco Nexus 3500 platform switches, the value of *pipeline-id* is 203.
Step 4	Required: **sub-switch** *sub-switch-id* **vlan** *vlan-range* **Example:** `Device(config-ofa-switch)#` **`sub-switch 2 vlan 301-305`**	Creates an OpenFlow logical subswitch for the specified VLAN or VLAN range. • The *sub-switch-id* is a unique ID for this sub-switch. It is an integer between 2 and 10. The primary switch has the ID of 1. • VLANs associated with this subswitch cannot also be associated to another subswitch, and VLAN ranges cannot overlap between subswitches. To return to the configuration of this subswitch later, you must repeat the exact command, including the subswitch ID and the VLAN range.

Step 5	**controller ipv4** *ip-address* [**port** *tcp-port*] [**vrf** *vrf-name*] **security**{**none** | **tls**}
	Example:
	Controller in default VRF:
	`Device(config-ofa-switch-subswitch)#`
	`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 that is 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, but the port is configurable to a different port number using the CLI.
		• You can configure up to eight controllers. Repeat this step if you need to configure more controllers.
		• If TLS is not disabled in this step, configure TLS trustpoints using the **tls** command.
		• You can use the **clear openflow switch 1 controller all** command to clear controller connections. This command can reset a connection after Transport Layer Security (TLS) certificates and keys are updated. This is not required for TCP connections.

	Command or Action	Purpose
		A connection to a controller is initiated for the logical switch.
Step 6	Required: **protocol-version** *version-info* **Example:** `Device(config-ofa-switch-subswitch)#` **`protocol-version 1.3`**	This command forces a specific version of the controller connection. If you force version 1.3 and the controller supports only 1.0, no session is established (or vice versa). The default behavior is to negotiate a compatible version between the controller and device. 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**—(Default) Negotiates the protocol version with the controller. Device uses 1.3 for negotiation.
Step 7	(Optional) **tls trust-point local** *local-trust-point* **remote** *remote-trust-point* **Example:** `Device(config-ofa-switch-subswitch)#` **`tls trust-point local mylocal remote myremote`**	Specifies the local and remote TLS trustpoints to be used for the controller connection. • For information on configuring trustpoints, refer to the "Configuring PKI" chapter of the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*.
Step 8	(Optional) **probe-interval** *probe-interval* **Example:** `Device(config-ofa-switch-subswitch)#` **`probe-interval 5`**	Configures the interval, in seconds, at which the controller is probed with echo requests. • The default value is 5. • The range is 5–65535.
Step 9	(Optional) **rate-limit packet_in** *controller-packet-rate* **burst** *maximum-packets-to-controller* **Example:** `Device(config-ofa-switch-subswitch)#` **`rate-limit packet_in 300 burst 50`**	Configures the maximum packet rate of the connection to the controller and the maximum packets that are permitted in a burst of packets that are 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 OpenFlow Agent. It is not related to the rate limit of the device (data plane) configured by CoPP.

	Command or Action	Purpose
Step 10	(Optional) **max-backoff** *backoff-timer* **Example:** `Device(config-ofa-switch-subswitch)# max-backoff 8`	Configures the time, in seconds, for which the device must wait before attempting to retry the connection with the controller. • The default value is eight. • The range is 1–65535 seconds.
Step 11	(Optional) **datapath-id** *id* **Example:** `Device(config-ofa-switch-subswitch)# datapath-id 0x111`	The identifier of the subswitch, which allows the controller to uniquely identify the device. This command overwrites the default value, which is based on the MAC address of the switch and the ID of the subswitch. A valid *id* is a 64-bit hex value in the range [0x1-0xffffffffffffffff].

Configuration Examples for Cisco OpenFlow Agent

Example: Enabling Cisco OpenFlow Agent in the Nexus 3000 series device

```
Device> enable
Device# configure terminal
Device(config)# system switch-mode n9k
Device# exit
Device# write erase
Device# reload
This command will reboot the system. (y/n)?  [n] y
.
.
.
[log in after reboot]
Device# configure terminal
Device(config)# feature openflow
Device(config)# show feature | inc openflow
openflow               1          enabled
```

Example: Enabling Cisco OpenFlow Agent in the Nexus 9000 series device

```
Device# configure terminal
Device(config)# feature openflow
Device(config)# show feature | inc openflow
openflow               1          enabled
```

Example: Adjusting the Number of Flow Entries

```
Device# configure terminal
Device(config)# hardware access-list tcam region racl 0
Device(config)# hardware access-list tcam region e-racl 0
Device(config)# hardware access-list tcam region l3qos 0
Device(config)# hardware access-list tcam region span 0
Device(config)# hardware access-list tcam region redirect 0
Device(config)# hardware access-list tcam region vpc-convergence 0
Device(config)# hardware access-list tcam region openflow 1024
Device(config)# exit
Device# copy running-config startup-config
Device# reload
```

Example: Configuring Global Variables for a Cisco OpenFlow Agent Logical Switch

```
Device# configure terminal
Device(config)# mac-learn disable
Device(config)# spanning-tree mode mst
Device(config)# vlan 2
Device(config-vlan)# end
```

Example: Configuring Control Plane Policing for Packets Sent to a Controller

```
Device# configure terminal
Device# setup


         ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes


  Create another login account (yes/no) [n]:

  Configure read-only SNMP community string (yes/no) [n]:

  Configure read-write SNMP community string (yes/no) [n]:

  Enter the switch name : QI32

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway? (yes/no) [y]: n

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: n

  Configure the ntp server? (yes/no) [n]:

  Configure default interface layer (L3/L2) [L2]:
```

```
  Configure default switchport interface state (shut/noshut) [noshut]:
  Configure CoPP System Policy Profile ( default / l2 / l3 ) [default]:

The following configuration will be applied:
  switchname QI32
  telnet server enable
  no ssh server enable
  system default switchport
  no system default switchport shutdown
  policy-map type control-plane copp-system-policy ( default )

Would you like to edit the configuration? (yes/no) [n]:

Use this configuration and save it? (yes/no) [y]:

[########################################] 100%
Copy complete, now saving to disk (please wait)...

Device# configure terminal
Device(config)# policy-map type control-plane copp-system-policy
Device(config-pmap)# class copp-s-dpss
Device(config-pmap-c)# police pps 1000
Device(config-pmap-c)# end
Device# show run copp
```

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

```
Device# configure terminal
Device(config)# interface ethernet1/1
Device(config-if)# no switchport
Device(config-if)# ip address 10.0.1.4 255.255.255.255
Device(config-if)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.1.6
Device# copy running-config startup-config
Device(config)# exit
```

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

```
Device# configure terminal
Device(config)# interface mgmt0
Device(config-if)# no switchport
Device(config-if)# ip address 10.0.1.4 255.255.255.255
Device(config-if)# exit
Device(config)# vrf context management
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.1.6
Device# copy running-config startup-config
Device(config)# exit
```

Example: Configuring an Interface for a Cisco OpenFlow Agent Logical Switch in L2 mode

```
Device# configure terminal

Device(config)# interface ethernet1/1
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# exit

Device(config)# interface ethernet1/2
```

```
! Adding the interface to a port channel.
Device(config-if)# channel-group 2
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
Device# copy running-config startup-config
```

Example: Configuring a Port-Channel Interface

```
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport mode trunk
Device(config-if)# end
Device# copy running-config startup-config
```

Example: Cisco OpenFlow Agent Logical Switch Configuration (Default VRF)

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 201
! Specifies the pipeline that enables the IP Forwarding Table.
Device(config-ofa-switch)# logging flow-mod
Device(config-ofa-switch)# max-backoff 5
Device(config-ofa-switch)# probe-interval 5
Device(config-ofa-switch)# rate-limit packet-in 300 burst 50
Device(config-ofa-switch)# controller ipv4 10.0.1.6 security none
! Adding an interface to the Cisco OpenFlow Agent logical switch.
Device(config-ofa-switch)# of-port interface ethernet1/1
Device(config-ofa-switch)# of-port interface ethernet1/2
! Adding a port channel to the Cisco OpenFlow Agent switch.
Device(config-ofa-switch)# of-port interface port-channel 2
Device(config-ofa-switch)# end
Device# copy running-config startup-config
```

Example: Configuring a Cisco OpenFlow Agent Logical Switch (Management VRF)

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 201
! Specifying a controller that is part of a VRF.
Device(config-ofa-switch)# controller ipv4 10.0.1.6 vrf management security none
! Adding an interface to the Cisco OpenFlow Agent logical switch.

Device(config-ofa-switch)# of-port interface ethernet1/1
Device(config-ofa-switch)# of-port interface ethernet1/2
! Adding a port channel to the Cisco OpenFlow Agent switch.
Device(config-ofa-switch)# of-port interface port-channel 2
Device(config-ofa-switch)# end
Device# copy running-config startup-config
```

Example: Creating a Sub-Switch

```
Device# configure terminal
Device(config)# openflow
Device(config-ofa)# switch 1 pipeline 201
Device(config-ofa-switch)# controller ipv4 5.30.199.200 port 6645 vrf management security
```

```
none
Device(config-ofa-switch)# of-port interface port-channel1000
Device(config-ofa-switch)# of-port interface Ethernet1/1
Device(config-ofa-switch)# of-port interface Ethernet1/37
Device(config-ofa-switch)# of-port interface Ethernet1/39
Device(config-ofa-switch)# logging flow-mod
Device(config-ofa-switch)# sub-switch 2 vlan 100
Device(config-ofa-switch-subswitch)# controller ipv4 5.30.19.239 port 6653 vrf management
security none
```

NXOS NXAPI Migration from OpenFlow

Beginning with Cisco Nexus Release 10.3(3)F, Openflow is not supported on N3500. Upgrade from older NXOS releases with Active Openflow configuration is not supported. Users should remove any active OpenFlow configuration on the device before the upgrade. After the upgrade, users can achieve similar redirect functionality using NXAPI.

Below are the steps to Updgrade Cisco Nexus 3500 devices and configure the same functionality through NXAPI.

Before you begin

Follow the steps as mentioned in Uninstalling Cisco Plug-in for OpenFlow and disable OpenFlow feature on NXOS.

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **no feature openflow**	Removes all openflow related configurations from the switch.
Step 2	switch(config)# **hardware profile forwarding-mode normal**	Sets hardware profile forwarding mode to normal.
Step 3	switch(config)# **no hardware profile openflow forward-pdu**	Reset openflow forward pdu flag in hardware. Ignore this step if you have not configured it. Save the configuration.

Achieving OpenFlow Functionality

Follow the steps as mentioned in **NXOS NXAPI Migration from OpenFlow** to upgrade to 10.3(3)F release. When the device is reloaded after upgrade, use the following commands to achieve OpenFlow functionality using CLIs supporting NXAPIs.

Before you begin

Follow the steps as mentioned in **NXOS NXAPI Migration from OpenFlow** to upgrade to 10.3(3)F release.

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **hardware profile tcam region ifacl-wide `4096/8192`**	Performs TCAM Carving. IFACL & IFACL-WIDE TCAM carvings are mutually exclusive, resize IFACL TCAM size to 0. This requires a box reload. Resize IFACL TCAM size to 0 first, before performing this step. Use the command **hardware profile tcam region ifacl 0** to reduce the ifacl tcam size. This requires a box reload.
Step 2	switch(config)# **hardware profile flow-redirect forward-pdu**	Enable forward pdu.
Step 3	switch(config)# **interface** *type number*	Configures an interface and enters interface configuration mode.
Step 4	switch(config-if)# **mode flow-redirect**	Redirects mode. Configure this on required trunk interfaces.

What to do next

See Configuring Wideflow IFACL Redirect on IP Port ACLs to define a new access-list with wideflow options. Attach IFACL to flow redirect interfaces as IFACL using standard ACL.

Note CLI support to define a new access-list Attach IFACL to flow redirect are interim till NDB starts supporting these. NDB GUI is recommended configuration approach once these options are supported by NDB.

Verifying Cisco OpenFlow Agent

Procedure

Step 1 **show openflow switch** *switch-id*

Displays information that is related to a Cisco OpenFlow Agent logical switch.

Example:

```
Device# show openflow switch 1

Logical Switch Context
  Id: 1
  Switch type: Forwarding
  Pipeline id: 201
  VLAN restrictions: none
  Data plane: secure
  Table-Miss default: controller
  Configured protocol version: Negotiate
  Config state: no-shutdown
  Working state: enabled
  Rate limit (packet per second): 300
  Burst limit: 50
  Max backoff (sec): 8
  Probe interval (sec): 5
  TLS local trustpoint name: not configured
  TLS remote trustpoint name: not configured
  Logging flow changes: Enabled
  Stats collect interval (sec): 7
  Stats collect Max flows: 3001
  Minimum flow idle timeout (sec): 14
  OFA Description:
    Manufacturer: Cisco Systems, Inc.
    Hardware: N9K-C9372PX 2.1
    Software: 7.0(3)I5(0.51)| of_agent 0.1
    Serial Num: SAL1944RZQN
    DP Description: switch:sw1
  OF Features:
    DPID: 0x0000000000009000
    Number of tables:1
    Number of buffers:256
    Capabilities: FLOW_STATS TABLE_STATS PORT_STATS
 Controllers:
    5.30.19.236:6653, Protocol: TCP, VRF: management
  Interfaces:
    Ethernet1/1
    Ethernet1/2
```

Step 2 **show openflow switch** *switch-id* **controllers** [**stats**]

Displays information that is related to the connection status between a Cisco OpenFlow Agent logical switch and connected controllers.

Example:

```
Device# show openflow switch 1 controllers

Logical Switch Id: 1
Total Controllers: 1
  Controller: 1
    5.30.19.236:6653
    Protocol: tcp
    VRF: management
    Connected: Yes
    Role: Master
    Negotiated Protocol Version: OpenFlow 1.3
    Last Alive Ping: 09/27/2016 00:04:53
    last_error:Connection timed out
    state:ACTIVE
    sec_since_connect:103334
    sec_since_disconnect:103345
    Current Role Since: 09/25/2016 19:22:41
```

The above sample output is displayed when the controller is connected (state:ACTIVE).

```
Device# show openflow switch 1 controllers stats

Logical Switch Id: 1
Total Controllers: 1
```

```
  Controller: 1
    address                         :  tcp:5.30.19.236:6653%management
    connection attempts             :  19
    successful connection attempts  :  2
    flow adds                       :  2
    flow mods                       :  0
    flow deletes                    :  0
    flow removals                   :  0
    flow errors                     :  0
    flow unencodable errors         :  0
    total errors                    :  0
    echo requests                   :  rx: 0, tx: 7
    echo reply                      :  rx: 6, tx: 0
    flow stats                      :  rx: 33763, tx: 33763
    barrier                         :  rx: 2, tx: 2
    packet-in/packet-out            :  rx: 0, tx: 23033
    Topology Monitor                :  rx: 0, tx: 0
    Topology State                  :  rx: 0
```

Step 3 **show running-config interface ethernet** *interface-id*

In the interface configuration, verify **mode openflow**.

Example:

```
Device# show running-config interface ethernet 1/2

!Command: show running-config interface Ethernet1/2
!Time: Thu Sep 29 00:08:18 2016

version 7.0(3)I5(1)

interface Ethernet1/7
no lldp transmit
spanning-tree bpdufilter enable
mode openflow
```

Step 4 **show openflow switch** *switch-id* **ports**

Displays the mapping between physical device interfaces and ports of a Cisco OpenFlow Agent logical switch.

Example:

```
Device# show openflow switch 1 ports

Logical Switch Id: 1
Port  Interface Name    Config-State   Link-State     Features
   2  Ethernet1/2       PORT_UP        LINK_UP        10MB-FD
   3  Ethernet1/3       PORT_UP        LINK_DOWN      100MB-HD AUTO_NEG
   4  Ethernet1/4       PORT_UP        LINK_UP        10MB-FD
```

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

Displays flows defined for the device by controllers.

Example:

```
Device# show openflow switch 1 flows

Logical Switch Id: 1
Total flows: 2
```

```
Flow: 1
  Match:              any
  Actions:            CONTROLLER:0
  Priority:           0
  Table:              0
  Cookie:             0x0
  Duration:           104160.376s
  Number of packets: 0
  Number of bytes:   0

Flow: 2
  Match:              in_port=2,dl_vlan=100
  Actions:            drop
  Priority:           100
  Table:              0
  Cookie:             0x0
  Duration:           103753.162s
  Number of packets: 0
  Number of bytes:   0
```

The following example show flows installed by the OpenFlow agent:

```
Device# show openflow switch 1 flows configured

Logical Switch Id: 1
Total flows: 1

Flow: 1
  Match:              any
  Actions:            CONTROLLER:0
  Priority:           0
  Table:              0
  Cookie:             0x0
  Duration:           104180.584s
  Number of packets: 0
  Number of bytes:   0
```

The following example show flows installed from the controller:

```
Device# show openflow switch 1 flows controller

Logical Switch Id: 1
Total flows: 1
Flow: 1
Match: in_port=2,dl_vlan=100
Actions: drop
Priority: 100
Table: 0
Cookie: 0x0
Duration: 103753.162s
Number of packets: 0
Number of bytes: 0
```

The following example displays the flow summary:

```
switch# show openflow switch 1 flows summary
Logical Switch Id: 1
Switch flow count: 2
```

The following example displays the brief version:

```
switch# show openflow switch 1 flows brief
Logical Switch Id: 1
Total flows: 3

Flow: 1 Match: any Actions: drop
Priority: 0, Table: 0, Cookie: 0x0, Duration: 127.349s, Packets: 7653260179, Bytes:
489808651630

Flow: 2 Match: dl_type=0x88cc Actions: CONTROLLER:65535
Priority: 50000, Table: 0, Cookie: 0x0, Duration: 127.431s, Packets: 14, Bytes: 1472

Flow: 3 Match: in_port=34,dl_type=0x800 Actions: output:20
Priority: 500, Table: 0, Cookie: 0x0, Duration: 127.432s, Packets: 63, Bytes: 4032
```

Step 6 **show openflow switch** *switch-id* **flow stats**

Displays send and receive statistics for each port that is defined for a Cisco OpenFlow Agent logical switch.

Example:

```
Device# show openflow switch 1 flow stats

Logical Switch Id: 1

Total ports: 2
  Port  1: rx pkts=96932, bytes=10911299, drop=0, errs=0,
           tx pkts=209683, bytes=19045035, drop=0, errs=0,
  Port  2: rx pkts=350485253, bytes=23834112937, drop=0, errs=0,
           tx pkts=191127, bytes=16001929, drop=0, errs=0,
Total tables: 1
  Table 0: NXOS PLCMGR IPV6 - PIPE 201
  Wildcards = 0x300033
  Max entries =   3001
  Active entries = 2
  Number of lookups = 0
  Number of matches = 0
```

Flow statistics are available for pipeline 201 and table 0. For pipeline 202, flow statistics are not available for table 1.

Step 7 **show logging last** *number-of-lines*

Displays logging information of flow changes, including addition, deletion, or modification of flows.

Example:

```
Device# show logging last 10

2016 Oct  5 09:52:27 switch of_agent: <{of_agent}> libpolicyshim:
policy_shim_parse_plcmgr_policy_stats 65
15 cmd_attr 352256118
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}> libpolicyshim:
policy_shim_parse_plcmgr_policy_stats 65
43 ppf_id 87032089
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}> libpolicyshim:
policy_shim_parse_plcmgr_policy_stats 65
15 cmd_attr 352256200
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}> libpolicyshim:
policy_shim_parse_plcmgr_policy_stats 65
36 pkts 0x9d3b bytes 0x0
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}>|-|00353|plif_xos_util|DBG|cstat
classified.pkts = 40251
```

```
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}>|-|00354|plif_xos_util|DBG|cstat
classified.bytes = 0
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}>|-|00355|plif_xos_util|DBG|cstat drop.pkts
 = 0
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}>|-|00356|plif_xos_util|DBG|cstat drop.bytes
 = 0
2016 Oct  5 09:52:27 switch of_agent: <{of_agent}>|-|00357|plif_xos|DBG|PXOS lookup switch
 by ls_id: switc
h ls_id is 1, passed in ls_id is 1
2016 Oct  5 09:52:28 switch of_agent: <{of_agent}>|-|1841673|poll_loop|DBG|wakeup due to
999-ms timeout at
 ../feature/sdn/openflow/cmn/ovs/cof_ovs_ofproto_plif.c:815 (0% CPU usage)
```

Step 8 **show running-config openflow**

Displays configurations that are made for Cisco OpenFlow Agent.

Example:

```
Device# show running-config openflow

!Command: show running-config openflow
!Time: Tue Sep 27 00:19:00 2016

version 7.0(3)I5(1)
feature openflow

openflow
  switch 1 pipeline 201
    rate-limit packet_in 300 burst 50
    probe-interval 5
    statistics collection-interval 7
    datapath-id 0x9000
    controller ipv4 5.30.19.236 port 6653 vrf management security none
    of-port interface Ethernet1/1
    of-port interface Ethernet1/2
    default-miss controller
    logging flow-mod
```

Step 9 **show running-config openflow**

Displays configurations that are made for Cisco OpenFlow Agent for Cisco Nexus 3500 platform switches.

Example:

```
Device# show running-config openflow

!Command: show running-config openflow
!Time: Tue Sep 27 00:19:00 2016

version 7.0(3)I7(8)
feature openflow

openflow
  switch 1 pipeline 203
    rate-limit packet_in 300 burst 50
    probe-interval 10
    max-backoff 5
    statistics collection-interval 7
    datapath-id 0x1
    controller ipv4 5.30.19.236 port 6653 vrf management security none
    of-port interface Ethernet1/17
    of-port interface Ethernet1/18
    of-port interface Ethernet1/19
    of-port interface Ethernet1/33
    of-port interface Ethernet1/48
    default-miss controller
    logging flow-mod
```

Step 10 **show openflow hardware capabilities**

Displays hardware capabilities for OpenFlow.

Example:

```
Device# show openflow hardware capabilities

  Max Interfaces: 1000
  Aggregated Statistics: NO

  Pipeline ID: 201
    Pipeline Max Flows: 3001
    Max Flow Batch Size: 300
    Statistics Max Polling Rate (flows/sec): 1024
    Pipeline Default Statistics Collect Interval: 7

    Flow table ID: 0

    Max Flow Batch Size: 300
    Max Flows: 3001
    Bind Subintfs: FALSE
    Primary Table: TRUE
    Table Programmable: TRUE
    Miss Programmable: TRUE
    Number of goto tables: 0
    goto table id:
    Stats collection time for full table (sec): 3

    Match Capabilities                    Match Types
    ------------------                    -----------
    ethernet type                         optional
    VLAN ID                               optional
    VLAN priority code point              optional
    IP DSCP                               optional
    IP protocol                           optional
    ipv6 source addresss                  lengthmask
    ipv6 destination address              lengthmask
    source port                           optional
    destination port                      optional
    in port (virtual or physical)         optional
    wildcard all matches                  optional

    Actions                        Count Limit        Order
    specified interface                     64            20
    controller                               1            20
    divert a copy of pkt to application      1            20

    set eth source mac                       1            10
    set eth destination mac                  1            10
    set vlan id                              1            10

    pop vlan tag                             1            10

    drop packet                              1            20


    Miss actions                   Count Limit        Order
    use normal forwarding                    1             0
    controller                               1            20

    drop packet                              1            20



  Max Interfaces: 1000
  Aggregated Statistics: NO

  Pipeline ID: 202
    Pipeline Max Flows: 3001
    Max Flow Batch Size: 300
    Statistics Max Polling Rate (flows/sec): 1024
    Pipeline Default Statistics Collect Interval: 7

    Flow table ID: 0

    Max Flow Batch Size: 300
    Max Flows: 3001
    Bind Subintfs: FALSE
    Primary Table: TRUE
    Table Programmable: TRUE
    Miss Programmable: TRUE
    Number of goto tables: 1
    goto table id:     1
    Stats collection time for full table (sec): 3

    Match Capabilities                    Match Types
    ------------------                    -----------
    ethernet type                         optional
    VLAN ID                               optional
    VLAN priority code point              optional
    IP DSCP                               optional
    IP protocol                           optional
    ipv6 source addresss                  lengthmask
    ipv6 destination address              lengthmask
    source port                           optional
    destination port                      optional
    in port (virtual or physical)         optional
    wildcard all matches                  optional

    Actions                             Count Limit        Order
    specified interface                           64            20
    controller                                     1            20
    divert a copy of pkt to application            1            20

    set eth source mac                             1            10
    set eth destination mac                        1            10
    set vlan id                                    1            10

    pop vlan tag                                   1            10

    drop packet                                    1            20


    Miss actions                        Count Limit        Order
    use normal forwarding                          1             0
    controller                                     1            20
    perform another lookup in the specified table 1            20

    drop packet                                    1            20
```

```
    Flow table ID: 1

    Max Flow Batch Size: 300
    Max Flows: 32001
    Bind Subintfs: FALSE
    Primary Table: FALSE
    Table Programmable: TRUE
    Miss Programmable: TRUE
    Number of goto tables: 0
    goto table id:
    Stats collection: Not Supported

    Match Capabilities                    Match Types
    ------------------                    -----------
    ethernet mac destination              mandatory
    VLAN ID                               mandatory
    wildcard all matches                  mandatory

    Actions                        Count Limit        Order
    specified interface                     64           20

    drop packet                              1           20


    Miss actions                   Count Limit        Order
    use normal forwarding                    1             0
    controller                               1            20

    drop packet                              1            20
```

Step 11 **show openflow switch 2**

Displays configuration of OpenFlow subswitch.

Example:

```
Device# show openflow switch 2

Logical Switch Context
  Id: 2
  Switch type: Forwarding
  Pipeline id: 201
  VLAN restrictions: 100
  Data plane: secure
  Table-Miss default: drop
  Configured protocol version: Negotiate
  Config state: no-shutdown
  Working state: enabled
  Rate limit (packet per second): 0
  Burst limit: 0
  Max backoff (sec): 8
  Probe interval (sec): 180
  TLS local trustpoint name: not configured
  TLS remote trustpoint name: not configured
  Logging flow changes: Disabled
  Stats collect interval (sec): 7
  Stats collect Max flows: 3001
  Minimum flow idle timeout (sec): 14
  OFA Description:
    Manufacturer: Cisco Systems, Inc.
    Hardware: N9K-C9372PX 2.1
    Software: 7.0(3)I5(0.51)| of_agent 0.1
    Serial Num: SAL1944RZQN
    DP Description: switch:sw2
  OF Features:
    DPID: 0x000258ac786b5457
    Number of tables:1
    Number of buffers:256
    Capabilities: FLOW_STATS TABLE_STATS PORT_STATS
 Controllers:
    5.30.19.239:6653, Protocol: TCP, VRF: management
  Interfaces:
    port-channel1000
    Ethernet1/1
    Ethernet1/37
    Ethernet1/39
```

Step 12 **show openflow switch 1**

Displays configuration of OpenFlow subswitch for Cisco Nexus 9500 platform switches.

Example:

```
Device# show openflow switch 1

Logical Switch Context
  Id: 1
  Switch type: Forwarding
  Pipeline id: 203
  VLAN restrictions: none
  Data plane: secure
  Table-Miss default: drop
  Configured protocol version: Negotiate
  Config state: no-shutdown
  Working state: enabled
  Rate limit (packet per second): 0
  Burst limit: 0
  Max backoff (sec): 5
  Probe interval (sec): 10
  TLS local trustpoint name: not configured
  TLS remote trustpoint name: not configured
  Logging flow changes: Enabled
  Stats collect interval (sec): 7
  Stats collect Max flows: 4095
  Minimum flow idle timeout (sec): 14
  OFA Description:
    Manufacturer: Cisco Systems, Inc.
    Hardware: N9K-C3548P-10G V00
    Software: 7.0(3)I7(8)| of_agent 0.1
    Serial Num: FOC163R04W
    DP Description: OF-MTC:sw1
  OF Features:
    DPID: 0x0001<>
    Number of tables:1
    Number of buffers:256
    Capabilities: FLOW_STATS TABLE_STATS PORT_STATS
    Actions: OUTPUT SET_VLAN_VID STRIP_VLAN
 Controllers:
    <>:6653, Protocol: TCP, VRF: management
  Interfaces:
    Ethernet1/17
    Ethernet1/18
    Ethernet1/19
    Ethernet1/33
    Ethernet1/48
```

Step 13 **show openflow switch 2 controllers stats**

Displays information that is related to the controller statistics for a logical subswitch.

Example:

```
Device# show openflow switch 2 controllers stats

Logical Switch Id: 2
Total Controllers: 1
  Controller: 1
    address                         :  tcp:5.30.19.239:6653%management
    connection attempts             :  5
    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
    Topology Monitor                :  rx: 0, tx: 0
    Topology State                  :  rx: 0
```

Step 14 **show run openflow**

Displays configurations that are made for Cisco OpenFlow Agent when a subswitch is configured.

Example:

```
Device# show run openflow

!Command: show running-config openflow
!Time: Thu Sep 29 00:09:21 2016

version 7.0(3)I5(1)
feature openflow

openflow
  switch 1 pipeline 201
    controller ipv4 5.30.199.200 port 6645 vrf management security none
    of-port interface port-channel1000
    of-port interface Ethernet1/1
    of-port interface Ethernet1/37
    of-port interface Ethernet1/39
    logging flow-mod
    sub-switch 2 vlan 100
      controller ipv4 5.30.19.239 port 6653 vrf management security none
```

Step 15 **show openflow hardware capabilities**

Displays configurations that are made for Cisco OpenFlow Agent when a subswitch is configured for Cisco Nexus 3500 platform switches.

Example:

```
Device# show openflow hardware capabilities

  Max Interfaces: 1000
  Aggregated Statistics: YES

  Pipeline ID: 203
    Pipeline Max Flows: 4095
    Max Flow Batch Size: 100
    Statistics Max Polling Rate (flows/sec): 1024
    Pipeline Default Statistics Collect Interval: 7

    Flow table ID: 0

    Max Flow Batch Size: 0
    Max Flows: 4095
    Bind Subintfs: FALSE
    Primary Table: TRUE
    Table Programmable: TRUE
    Miss Programmable: TRUE
    Number of goto tables: 0
    goto table id:
    Stats collection time for full table (sec): 4

    Match Capabilities                            Match Types
    ------------------                            -----------
    ethernet mac destination                            optional
    ethernet mac source                                 optional
    ethernet type                                       optional
    VLAN ID                                             optional
    IP DSCP                                             optional
    IP protocol                                         optional
    IPv4 source address                                 lengthmask
    IPv4 destination address                            lengthmask
    source port                                         optional
    destination port                                    optional
    in port (virtual or physical)                       optional

    Actions                                 Count Limit      Order
    specified interface                                 64                    20
    controller                                           1                    20

    set vlan id                                          1                    10

    pop vlan tag                                         1                    10

    drop packet                                          1                    20


    Miss actions                       Count Limit           Order
    specified interface                                 64                    20
    controller                                           1                    20

    drop packet                                          1                    20
```

Additional Information for Cisco OpenFlow Agent

Related Documents

Related Topic	Document Title
Cisco command references	Cisco Nexus 3000 Series Switches Command References Cisco Nexus 9000 Series Switches Command References

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 OpenFlow Agent

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Cisco OpenFlow Agent

Feature Name	Releases	Feature Information
Cisco OpenFlow Agent	7.0(3)I5(1)	Cisco OpenFlow Agent is introduced, replacing the Cisco Plug-in for OpenFlow used in previous NX-OS releases.

APPENDIX A

Supported Platforms for Cisco OpenFlow Agent

Supported Platforms for Cisco OpenFlow Agent

Nexus 3000 Series

Platform	OpenFlow Support (Pipeline Number)
Cisco Nexus 30* Switch	201/202
Cisco Nexus 3132/3172* Switch	201/202
Cisco Nexus 3132QV Switch	201/202
Cisco Nexus 31108PCV Switch	201/202
Cisco Nexus 31108TCV Switch	201/202
Cisco Nexus 31128PQ-10GE Switch	201/202
Cisco Nexus 3232C Switch	201/202
Cisco Nexus 3264Q Switch	201/202
Cisco Nexus3000 C3164PQ Chassis	No
Cisco Nexus 3548 switch	203

Nexus 9000 Series

Platform	OpenFlow Support (Pipeline Number)
Cisco Nexus 9332PQ Switch	201/202
Cisco Nexus 9372PX Switch	201/202
Cisco Nexus 9372TX Switch	201/202

Platform	OpenFlow Support (Pipeline Number)
Cisco Nexus 9396PX Switch	201/202 201/202
Cisco Nexus 9396TX Switch	201/202
Cisco Nexus 93120TX Switch	201/202
Cisco Nexus 93128TX Switch	201/202
Cisco Nexus 9504 Switch	205 (see note)
Cisco Nexus 9508 Switch	205 (see note)
Cisco Nexus 9516 Switch	205 (see note)

Note OpenFlow pipeline 205 is supported on Cisco Nexus 95XX switches only when the switch contains the Application Spine Engine 2 (ASE2), Application Spine Engine 3 (ASE3), or Leaf Spine Engine (LSE). If any fabric board other than these is present, the OpenFlow feature cannot be enabled.

APPENDIX B

Uninstalling Cisco Plug-in for OpenFlow

Uninstalling Cisco Plug-in for OpenFlow

The Cisco OpenFlow Agent was introduced in Cisco NX-OS Release 7.0(3)I5(1), replacing the Cisco Plug-in for OpenFlow used in previous releases. The Cisco Plug-in for OpenFlow, which runs as an application in a virtual services container, is no longer supported as of this release. When upgrading from a release earlier than Cisco NX-OS Release 7.0(3)I5(1) to Cisco NX-OS Release 7.0(3)I5(1) or a later release, you must deactivate and uninstall the Cisco Plug-in for OpenFlow application from the virtual services container using the procedure described in this section.

Cisco OpenFlow Agent support for the Cisco Nexus 3548 was introduced in Cisco NX-OS Release 7.0(3)I7(2) replacing the Cisco Plug-in for OpenFlow used from Cisco NX-OS Release 6.0(2)A8(1). When upgrading form a release earlier than Cisco NX-OS Release 7.0(3)I7(2) to Cisco NX-OS Release 7.0(3)I7(2) or a later release, you must deactivate and uninstall the Cisco Plug-in for OpenFlow application from the virtual services container using the procedure described in this section.

Converting a Previous OpenFlow Configuration

When you upgrade to a release that requires you to uninstall the Cisco Plug-in for OpenFlow, you can save your existing OpenFlow configuration and modify it for use with the Cisco OpenFlow Agent. Perform the following procedure before uninstalling the Cisco Plug-in for OpenFlow.

Procedure

Step 1 Capture the current OpenFlow configuration.

Enter the CLI command **show run | section openflow** to display the current OpenFlow configuration, as shown in this example.

Example:

```
Switch# show run | section openflow
```

```
hardware access-list tcam region openflow 512 double-wide
  mode openflow
  mode openflow
  mode openflow
  mode openflow
  mode openflow
openflow            <-----------  Copy this section to your text editor.
  switch 1
    pipeline 201
    controller ipv4 5.1.1.237 port 6653 vrf management security none
    of-port interface Ethernet1/11-15
```

Step 2 Copy the configuration and paste it into your text editor.

Step 3 Make the changes described below.

- Add the **feature openflow** command to enable the Cisco OpenFlow Agent.
- Combine the **switch** and **pipeline** commands into one command.
- Expand any interface ranges.

Example:

```
feature openflow  <------------------------- Add this comment to enable openflow agent

openflow
  switch 1 pipeline 201  <------------------ Create switch command is in this format
    controller ipv4 192.168.1.36 port 6653
    of-port interface Ethernet1/11  <------- Change Ethernet1/11-15 to this format
    of-port interface Ethernet1/12
    of-port interface Ethernet1/13
    of-port interface Ethernet1/14
    of-port interface Ethernet1/15
```

Step 4 Make the changes described below.

When upgrading from a release earlier than Cisco NX-OS Release 7.0(3)I7(2) to Cisco NX-OS Release 7.0(3)I7(2) or a later release, obtain the Node-ID of the switch from the OpenFlow controller to which the switch is registered. Copy the Node-ID in your text editor. After upgrading, configure the Node-ID under OpenFlow.

Example:

```
openflow
switch 1 pipeline 201
controller ipv4 192.168.1.36 port 6653 vfr management security none
datapath-id 0x174a02fc67f00
```

Note 0x174a02fc67f00 is the Node-ID of the switch which had been registered with the OpenFlow controller before upgrading.

Step 5 Make the changes described below.

If the 'onep_apps_openflow_GLOBAL_VER.cli' file exists under bootflash:onep/apps-cli, it must be removed if you are upgrading from a release earlier than Cisco NX-OS Release 7.0(3)I7(2) to Cisco NX-OS Release 7.0(3)I7(2) or a later release.

Example:

```
switch# delete bootflash:onep/apps-cli/onep_apps_openflow_GLOBAL_VER.cli
```

What to do next

After uninstalling the Cisco Plug-in for OpenFlow, uninstalling the virtual service container (if necessary), and upgrading the switch, follow the instructions in this guide to enable the Cisco OpenFlow Agent. Then load the modified configuration into the switch.

Deactivating and Uninstalling an Application from a Virtual Services Container

(Optional) Perform this task to uninstall and deactivate an application from within a virtual services container.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**virtual-service** *virtual-services-name* **Example:** `Device(config)# virtual-service openflow_agent`	Enters virtual services configuration mode to configure a specified application. • Use the *virtual-services-name* defined during installation of the application.
Step 4	**no activate** **Example:** `Device(config-virt-serv)# no activate`	Disables the application.
Step 5	**no virtual-service** *virtual-services-name* **Example:** `Device(config)# no virtual-service openflow_agent`	Unprovisions the application. • Use the *virtual-services-name* defined during installation of the application. • This command is optional for all devices running Cisco IOS-XE.

Step 6	**end**
	Example:
	`Device(config-virt-serv)# end`
		Exits virtual services configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 7	**virtual-service uninstall name** *virtual-services-name* **Example:** `Device# virtual-service uninstall name openflow_agent`	Uninstalls the application. • Use the *virtual-services-name* defined during installation of the application. • Run this command only after receiving a successful deactivation response from the device.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.