CONFIGURATION GUIDE

RUCKUS FastIron SDN Configuration Guide, 09.0.10

Supporting FastIron Software Release 09.0.10

Part Number: 53-1005717-01
Publication Date: 2 December 2021

Copyright, Trademark and Proprietary Rights Information

Export Restrictions

These products and associated technical data (in print or electronic form) may be subject to export control laws of the United States of America. It is your responsibility to determine the applicable regulations and to comply with them. The following notice is applicable for all products or technology subject to export control:

These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. government or as otherwise authorized by U.S. law and regulations.

Disclaimer

THIS CONTENT AND ASSOCIATED PRODUCTS OR SERVICES ("MATERIALS"), ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT PERMISSIBLE PURSUANT TO APPLICABLE LAW, COMMSCOPE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, FREEDOM FROM COMPUTER VIRUS, AND WARRANTIES ARISING FROM COURSE OF DEALING OR COURSE OF PERFORMANCE. CommScope does not represent or warrant that the functions described or contained in the Materials will be uninterrupted or error-free, that defects will be corrected, or are free of viruses or other harmful components. CommScope does not make any warranties or representations regarding the use of the Materials in terms of their completeness, correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise. As a condition of your use of the Materials, you warrant to CommScope that you will not make use thereof for any purpose that is unlawful or prohibited by their associated terms of use.

Limitation of Liability

IN NO EVENT SHALL COMMSCOPE, COMMSCOPE AFFILIATES, OR THEIR OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, SUPPLIERS, LICENSORS AND THIRD PARTY PARTNERS, BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER, EVEN IF COMMSCOPE HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT, OR ANY OTHER THEORY ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIALS. Because some jurisdictions do not allow limitations on how long an implied warranty lasts, or the exclusion or limitation of liability for consequential or incidental damages, some of the above limitations may not apply to you.

Trademarks

ARRIS, the ARRIS logo, COMMSCOPE, RUCKUS, RUCKUS WIRELESS, the Ruckus logo, the Big Dog design, BEAMFLEX, CHANNELFLY, FASTIRON, ICX, SMARTCELL and UNLEASHED are trademarks of CommScope, Inc. and/or its affiliates. Wi-Fi Alliance, Wi-Fi, the Wi-Fi logo, Wi-Fi Certified, the Wi-Fi CERTIFIED logo, Wi-Fi Protected Access, the Wi-Fi Protected Setup logo, Wi-Fi Protected Setup, Wi-Fi Multimedia and WPA2 and WMM are trademarks or registered trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

Contents

Preface

Contacting RUCKUS Customer Services and Support

The Customer Services and Support (CSS) organization is available to provide assistance to customers with active warranties on their RUCKUS products, and customers and partners with active support contracts.

For product support information and details on contacting the Support Team, go directly to the RUCKUS Support Portal using https://support.ruckuswireless.com, or go to https://www.ruckuswireless.com and select **Support**.

What Support Do I Need?

Technical issues are usually described in terms of priority (or severity). To determine if you need to call and open a case or access the self-service resources, use the following criteria:

- Priority 1 (P1)—Critical. Network or service is down and business is impacted. No known workaround. Go to the **Open a Case** section.
- Priority 2 (P2)—High. Network or service is impacted, but not down. Business impact may be high. Workaround may be available. Go to the **Open a Case** section.
- Priority 3 (P3)—Medium. Network or service is moderately impacted, but most business remains functional. Go to the **Self-Service Resources** section.
- Priority 4 (P4)—Low. Requests for information, product documentation, or product enhancements. Go to the **Self-Service Resources** section.

Open a Case

When your entire network is down (P1), or severely impacted (P2), call the appropriate telephone number listed below to get help:

- Continental United States: 1-855-782-5871
- Canada: 1-855-782-5871
- Europe, Middle East, Africa, Central and South America, and Asia Pacific, toll-free numbers are available at https://support.ruckuswireless.com/contact-us and Live Chat is also available.
- Worldwide toll number for our support organization. Phone charges will apply: +1-650-265-0903

We suggest that you keep a physical note of the appropriate support number in case you have an entire network outage.

Self-Service Resources

The RUCKUS Support Portal at https://support.ruckuswireless.com offers a number of tools to help you to research and resolve problems with your RUCKUS products, including:

- Technical Documentation—https://support.ruckuswireless.com/documents
- Community Forums—https://forums.ruckuswireless.com/ruckuswireless/categories
- Knowledge Base Articles—https://support.ruckuswireless.com/answers
- Software Downloads and Release Notes—https://support.ruckuswireless.com/#products_grid
- Security Bulletins—https://support.ruckuswireless.com/security

Using these resources will help you to resolve some issues, and will provide TAC with additional data from your troubleshooting analysis if you still require assistance through a support case or RMA. If you still require help, open and manage your case at https://support.ruckuswireless.com/case_management.

Document Feedback

RUCKUS is interested in improving its documentation and welcomes your comments and suggestions.

You can email your comments to RUCKUS at #Ruckus-Docs@commscope.com.

When contacting us, include the following information:

- Document title and release number
- Document part number (on the cover page)
- Page number (if appropriate)

For example:

- RUCKUS SmartZone Upgrade Guide, Release 5.0
- Part number: 800-71850-001 Rev A
- Page 7

RUCKUS Product Documentation Resources

Visit the RUCKUS website to locate related documentation for your product and additional RUCKUS resources.

Release Notes and other user documentation are available at https://support.ruckuswireless.com/documents. You can locate the documentation by product or perform a text search. Access to Release Notes requires an active support contract and a RUCKUS Support Portal user account. Other technical documentation content is available without logging in to the RUCKUS Support Portal.

White papers, data sheets, and other product documentation are available at https://www.ruckuswireless.com.

Online Training Resources

To access a variety of online RUCKUS training modules, including free introductory courses to wireless networking essentials, site surveys, and products, visit the RUCKUS Training Portal at https://commscopeuniversity.myabsorb.com/.

Document Conventions

The following table lists the text conventions that are used throughout this guide.

TABLE 1 Text Conventions

Convention	Description	Example
`monospace`	Identifies command syntax examples	`device(config)# interface ethernet 1/1/6`
bold	User interface (UI) components such as screen or page names, keyboard keys, software buttons, and field names	On the **Start** menu, click **All Programs**.
italics	Publication titles	Refer to the *RUCKUS Small Cell Release Notes* for more information.

Notes, Cautions, and Safety Warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A NOTE provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An ATTENTION statement indicates some information that you must read before continuing with the current action or task.

CAUTION
A CAUTION statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A DANGER statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Command Syntax Conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{**x**| **y**| **z**}	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x|**y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

About This Document

Supported Hardware

This guide supports the following RUCKUS products:

- RUCKUS ICX 7850 Switch
- RUCKUS ICX 7650 Switch
- RUCKUS ICX 7550 Switch
- RUCKUS ICX 7450 Switch
- RUCKUS ICX 7250 Switch
- RUCKUS ICX 7150 Switch

For information about what models and modules these devices support, refer to the hardware installation guide for the specific product family.

What's new in this document

The following table describes changes to this guide for the FastIron 09.0.10 release.

TABLE 2 Summary of Changes in FastIron Release 09.0.10

Feature	Description	Reference
Minor editorial updates	Editorial updates and corrections were made.	Throughout the guide.

How Command Information is Presented in this Configuration Guide

For all new content supported in FastIron release 08.0.20 and later, command information is documented in a standalone command reference guide.

In the *RUCKUS FastIron Command Reference*, the command pages are in alphabetical order and follow a standard format to present syntax, parameters, mode, usage guidelines, examples, and command history.

NOTE

Many commands introduced before FastIron release 08.0.20 are also included in the guide.

OpenFlow v1.0.0

Overview of OpenFlow v1.0.0

An OpenFlow-enabled router supports an OpenFlow Client (control plane software), which communicates with an OpenFlow Controller using OpenFlow. The OpenFlow Controller runs on a server or a server cluster. OpenFlow-enabled routers support the abstraction of a flow table, which is manipulated by the OpenFlow Controller. The flow table contains flow entries. Each flow entry represents a flow (that is, packets with a given MAC address, VLAN tag, IP address, or TCP/UDP port, and so on). The flow table is sorted by flow priority, which is defined by the OpenFlow Controller. The highest priority flows are at the top of the flow table.

Incoming packets on an OpenFlow-enabled port are matched (in order of priority) against the flow entries defined for that port by the OpenFlow Controller. If the packet matches a given flow entry, the flow-matching process stops, and the set of actions defined for that flow entry are performed. Packets that do not match any flow entry are dropped by default. The RUCKUS ICX implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 20.

FIGURE 1 OpenFlow-enabled router

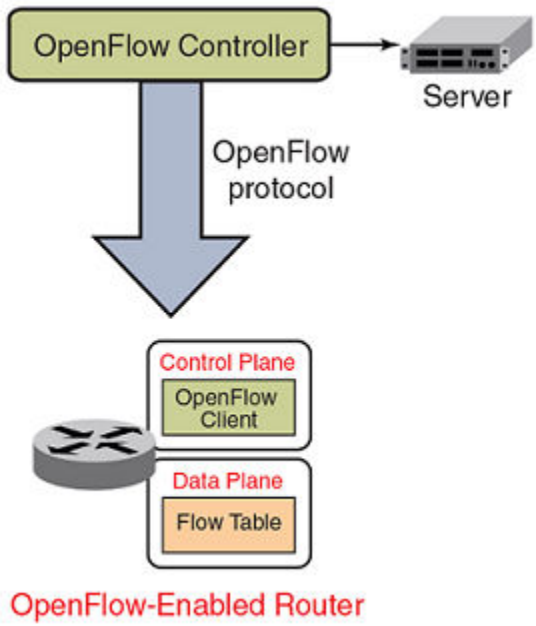
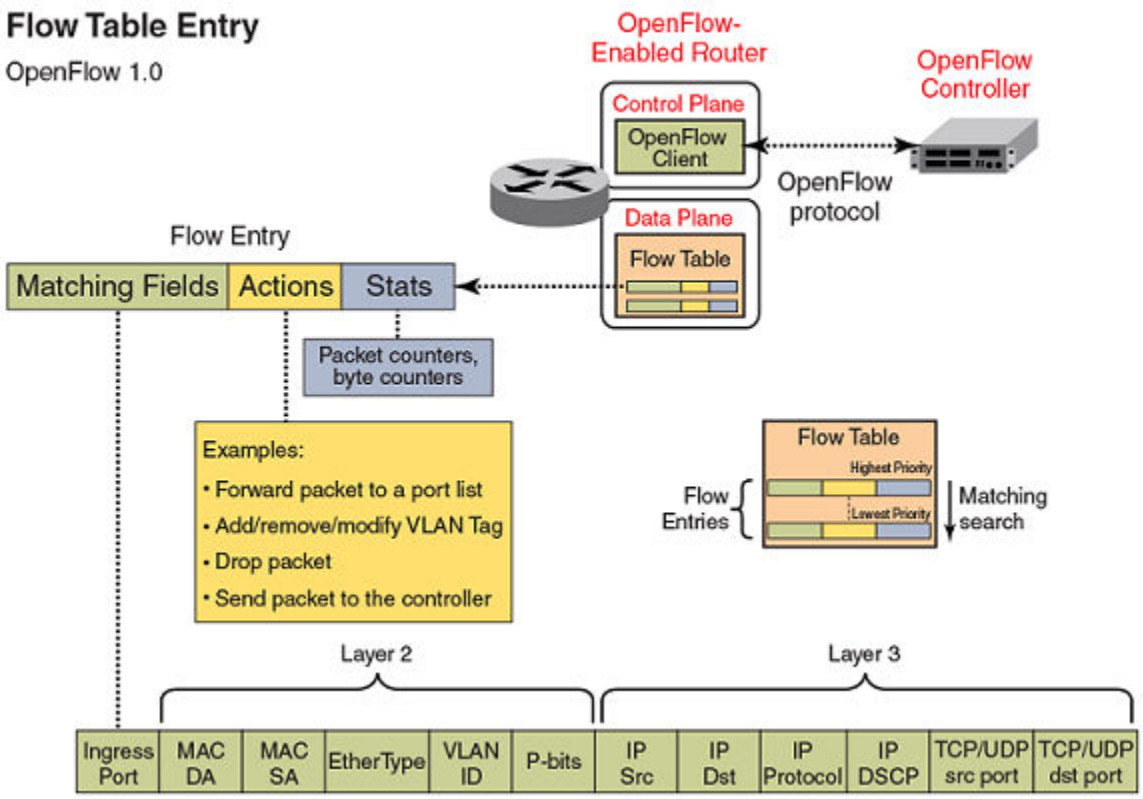


FIGURE 2 OpenFlow flow table entries



Flow table entries

The OpenFlow match rules in the following table are supported on RUCKUS ICX devices for flow table entries.

The implementation of OpenFlow supports three modes of operation when enabling OpenFlow on a port: Layer 2 mode, Layer 3 mode, and Layer23 mode. Layer 2 mode supports OpenFlow matching rules based on the Layer 2 fields shown in Overview of OpenFlow v1.0.0 on page 11, while Layer 3 mode supports the OpenFlow matching rules based on the Layer 3 fields. Layer23 mode supports the OpenFlow matching rules based on the Layer 2 and Layer 3 fields.

The RUCKUSICX 7650, ICX 7450, and ICX 7250 devices support enabling ports in either Layer 2, Layer 3, or Layer23 mode.

TABLE 3 OpenFlow Match Rules

Match rule	RUCKUS ICX device
Port enabled for Layer 2 mode	Yes
Source port	Yes
Source or destination MAC address	Yes These devices support either source MAC addres or destination MAC address, or a combination of both source and destination MAC addresses as the match rule.
Ether type	Yes
VLAN ID	Yes
VLAN priority	Yes
Port enabled for Layer 3 mode	Yes

TABLE 3 OpenFlow Match Rules (continued)

Match rule	RUCKUS ICX device
Ether type	No
Source port	Yes
VLAN ID	Yes
VLAN priority	Yes
Source IP address	Yes
Destination IP address	Yes
Protocol type	Yes
IP TOS bits	Yes
TCP or UDP source port	Yes
TCP or UDP destination port	Yes
Port enabled for Layer23 mode	Yes
Source port	Yes
Source or destination MAC address	Yes These devices support either source MAC addres or destination MAC address, or a combination of both source and destination MAC addresses as the match rule.
Ether type	Yes
VLAN ID	Yes
VLAN priority	Yes
Source IP address	No
Destination IP address	Yes
Protocol type	Yes
IP TOS bits	Yes
TCP or UDP source port	Yes
TCP or UDP destination port	Yes

OpenFlow actions

Each OpenFlow flow table entry contains the list of actions to be performed when a packet matches the flow entry. These actions are defined by the OpenFlow Controller.

Packets that do not match any flow entry are dropped by default. The RUCKUS ICX implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 20.

RUCKUS ICX devices support the actions listed in the following table.

TABLE 4 OpenFlow actions supported on RUCKUS ICX devices

OpenFlow action	RUCKUS ICX device
Forward a packet to a set of ports	Yes
Drop the packet	Yes
Add, modify, or remove VLAN ID or priority on a per -destination-port basis	Yes
Modify the IP DSCP (For a flow sending a copy of the packet to multiple destinations, the DSCP modification must be the same for all destinations. Modifying IP DSCP is only supported on ports enabled with Layer 3 mode.)	Yes
Modify the destination MAC address	Yes

TABLE 4 OpenFlow actions supported on RUCKUS ICX devices (continued)

OpenFlow action	RUCKUS ICX device
Send the packet to the OpenFlow Controller (Packet In)	Yes
Receive the packet from the OpenFlow Controller and send it to ports (Packet Out)	Yes

OpenFlow Controller

Multiple controller connections can be used for redundancy purposes, such as when using a single controller with multiple addresses. Multiple controller connections can also be used to support active-standby controllers.

Regardless of the intended use of multiple controller connections, the device allows all the controller connections to concurrently manage the flow table. That is, flow entries in the flow table are not identified as belonging to any specific controller connection. In an active-standby controller deployment, controllers themselves must coordinate their actions and active-standby states. The device responds to all connected controllers without distinction.

The device supports two types of controller connections (also called modes): active and passive. An active connection is one for which the device initiates (seeks) the TCP connection to a given OpenFlow Controller address. With a passive connection, the device passively waits for the controller to initiate (seek) the TCP connection to the device. Active mode is commonly used with production controllers, while passive mode is commonly used for testing purposes in experimental environments. Optionally, a controller connection can also use SSL encryption.

Considerations and limitations for configuring OpenFlow

Consider the following points when you configure OpenFlow on devices:

- OpenFlow must be enabled globally on the device before you can enable interfaces for OpenFlow.
- You must explicitly enable or disable OpenFlow on each interface using the CLI commands. You cannot use a range of ports to enable OpenFlow on the interface.
- Before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- Spanning Tree Protocol (STP) and other Layer 2 or Layer 3 protocols are not supported on OpenFlow-enabled ports.
- OpenFlow supports up to four concurrent sessions with a maximum of two concurrent SSL sessions.
- Layer 2 unicast and multicast packets are flooded in the VLAN for protected VLANs and for unprotected VLANs in absence of flows on hybrid OpenFlow ports.
- Local and normal actions defined by the OpenFlow v1.0.0 protocol are not supported.
- OpenFlow is an ingress feature. The local device generates protocol messages (such as PIM and OSPF) on OpenFlow enabled-ports, if configured, but control packets OpenFlow default rule. Because of this limitation, the PIM neighbor (if configured) comes up on the peer, and multicast traffic hits the OpenFlow interface in all PIM DMs. In a PIM SM, the OpenFlow port connects to an IGMP snooping-enabled LAN that has the multicast source connected.
- On OpenFlow-enabled ports, packets that do not match any flow entry are dropped by default. The RUCKUS ICX implementation supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 20.
- In openflow, the untagged frames arriving on hybrid interfaces will be treated as protected, that is they will be subjected to L3 processing and will not be subjected to openflow-flows.

OpenFlow hybrid switch mode and OpenFlow hybrid port mode

OpenFlow hybrid switch mode

The RUCKUS ICX device supports enabling OpenFlow on a per-port basis, so you can choose which ports of the device will be controlled by the OpenFlow feature. Non-OpenFlow-enabled ports continue to support existing features of the device, such as IPv4 or IPv6 routing for Layer 2 switching.

OpenFlow hybrid port mode

OpenFlow hybrid-enabled ports support both OpenFlow traffic forwarding and normal routing traffic forwarding. OpenFlow hybrid-enabled ports support "protected VLANs" and "unprotected VLANs". Protected VLANs are not subject to defined OpenFlow flows on the OpenFlow hybrid-enabled ports. OpenFlow flows on a hybrid-enabled port will not match any traffic on protected VLANs. Unprotected VLANs are subject to defined OpenFlow flows on the OpenFlow hybrid-enabled port. OpenFlow flows on a hybrid-enabled port are allowed to match on the traffic of unprotected VLANs.

NOTE
To set the flow, the VLAN id is must if the port is untagged and have unprotected VLAN. Without the VLAN id, error is shown and the flow installation cannot be done.

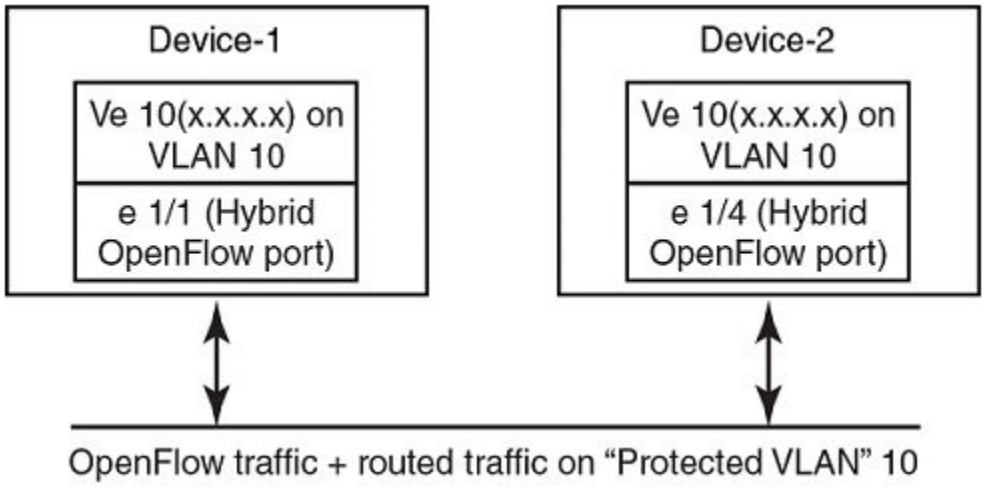
NOTE
The openflow L2 or L3 lookup does not work on hybrid interfaces for default VLAN.

NOTE
On the RUCKUS ICX 7650, the following restrictions apply:

- Ports with OpenFlow hybrid mode enabled cannot be added to an untagged VLAN group.
- OpenFlow hybrid mode cannot be enabled on ports added to an untagged VLAN group.

Figure 3 shows a topology in which port 1/1 on Device-1 and port 1/4 on Device-2 are hybrid-enabled OpenFlow ports with VLAN 10 as a configured protected VLAN. By configuring a virtual Ethernet (VE) interface on a protected VLAN 10 and assigning an address to route the traffic of the nodes, you are able to send protected VLAN traffic between the nodes and route the traffic as per the VE interface. Traffic flowing on other VEs created on top of other VLANs (the unprotected VLANs) is treated as unprotected VLAN traffic and is subject to OpenFlow rules lookup. OpenFlow traffic can be forwarded through this port.

FIGURE 3 OpenFlow hybrid port mode topology



OpenFlow hybrid port mode operation

Consider Device-1 in OpenFlow hybrid port mode on page 15. Ingress traffic on VLAN 10 on hybrid port 1/1 is processed for IPv4 and IPv6 unicast routing. Traffic on other VLANs is processed against OpenFlow flows on port 1/1 and switched accordingly. A preconfigured number of protected VLANs can be supported for normal routing. The Spanning Tree Protocol (STP) state of these routing VLANs is set to forwarding, as the Layer 2 protocol is not supported.

Configuring OpenFlow hybrid port mode for devices

1. Enable OpenFlow at the global configuration level.
2. Configure the OpenFlow Controller.
3. Configure the system maximum OpenFlow entries. (The default is 2048.)
4. Configure the maximum OpenFlow flow-protected VLAN entries. (The default is 40.)

   NOTE
   System reload is required once you change the system maximum values.

5. Configure the maximum OpenFlow unprotected VLAN entries. (The default is 40.)
6. Configure the protected VLANs on the port. A maximum of 40 protected VLANs can be configured on an OpenFlow port.
7. Enable OpenFlow hybrid port mode on the desired interfaces.
8. Configure a VE interface for the interface by specifying the protected or unprotected VLAN, and add routing entries.

Capabilities and prerequisites

The following are current capabilities and prerequisites of OpenFlow hybrid port mode:

- IPv4 and IPv6 unicast routing are supported on OpenFlow protected and unprotected VLANs.
- Packets tagged with a protected VLAN ID are forwarded by IPv4 and IPv6 unicast routing, if IPv4 or IPv6 routing is configured on that VLAN. If IPv4 or IPv6 routing is not configured on that VLAN, such packets are dropped.

- Packets tagged with an unprotected VLAN ID are subject first to OpenFlow flows. If there is a match on an OpenFlow flow, the packet is forwarded according to the flow actions. No further IPv4 or IPv6 routing is supported for packets that are forwarded by OpenFlow flows. If there is no match on any OpenFlow flow, the packet is forwarded by IPv4 or IPv6 unicast routing, if IPv4 or IPv6 routing is configured on the VLAN. If IPv4 or IPv6 routing is not configured on the VLAN, those packets are either dropped or sent to the controller, per the OpenFlow configuration.
- As routing is enabled on a port in OpenFlow hybrid port mode, OpenFlow traffic or unprotected VLAN traffic sent with the destination MAC address as the port's MAC address and matching IP route entries on the port can potentially find the VLAN and MAC address modified unless the OpenFlow rules explicitly set the VLAN and destination MAC address in the outgoing packet.
- Policy-Based Routing (PBR) is not supported.
- Protected VLAN traffic that does not have matching IP route entries is dropped.
- Multiple interfaces cannot be part of a VE interface created on a port in OpenFlow hybrid port mode with a protected VLAN.
- The BGP4+, OSPFv2, OSPFv3, RIP, and RIPng protocols are supported on protected VLANs.
- When protected VLANs are configured but the port is not part of the VLAN, the traffic coming on the port with the protected VLAN is dropped.
- Link aggregation is not supported.

Enabling OpenFlow hybrid port mode

Use the **openflow enable** command to enable or disable OpenFlow hybrid port mode on the port and the port becomes a normal port on an interface. The **no** form of the command disables the OpenFlow hybrid port mode on the port and the port becomes a normal port.

```
device(config-if-e10000-1/2/5)# openflow enable layer2 hybrid-mode
```

Adding or deleting protected VLANs

Use the **openflow protected-vlans** command to add or delete protected VLANs on an OpenFlow hybrid port mode interface. The **no** form of the command deletes the configured protected VLANs from the hybrid-enabled port.

```
device(config-if-e10000-1/2/5)# openflow protected-vlans 10
```

VLANs can be configured individually.

NOTE
You cannot specify a VLAN range for the **openflow protected-vlans** command.

Configuring OpenFlow

You can enable OpenFlow on an interface with Layer23 flows in order to support Layer 2 and Layer 3 flows on that interface. Layer23 flows support the OpenFlow hybrid port mode also. Configured with Layer23, the controller can configure flows with Layer 2 and Layer 3 parameters together. A flow can contain the following fields: Ingress port, Destination MAC address, Source MAC address, Ether type, VLAN ID , P-bits, Source IP address, Destination IP address, IP protocol, and IP DSCP.

NOTE
Source IP address is not supported on Layer23 flows.

By default, OpenFlow is disabled. You must first enable OpenFlow on the device before you can configure the parameters on the device.

Enabling OpenFlow on devices

After you enable OpenFlow on the device, you can enable OpenFlow on specific interfaces and configure additional OpenFlow parameters.

To enable OpenFlow, enter the following command:

```
device(config)# openflow enable ofv100
```

The **ofv100** keyword specifies the OpenFlow protocol version supported.

Use the **no** form of the command to disable OpenFlow on the device.

> **NOTE**
> You must disable OpenFlow on all interfaces individually before you can disable OpenFlow globally on the device.

Enabling OpenFlow on a specified interface

After you have enabled OpenFlow on the device, you can enable OpenFlow on specific interfaces.

> **NOTE**
> You can enable OpenFlow on an interface only after you have enabled OpenFlow globally on the device. In addition, you must use individual CLI commands to enable OpenFlow on each interface. You cannot specify a range of ports when enabling OpenFlow.

> **NOTE**
> Configuration of an OpenFlow hybrid port is not supported, if the port is already configured as a member of an MCT VLAN.

To enable OpenFlow on a specific interface, enter the following command:

```
device(config-if-e1000-1/1/1)# openflow enable layer2
```

You can specify Layer 2 or Layer 3 or both layers (as Layer23 matching mode) in OpenFlow hybrid port mode to be supported on the interface. By default, interfaces on these devices support Layer 2 matching mode. If you enable Layer 2 matching mode on the specified interface, only Layer 2 matching fields are supported on that interface.

Use the **no** form of the command to disable OpenFlow on the interface.

Flow validation

The following validations are required before programming flows on a Layer23 port:

- When IP fields exist in rule, then the Ether type must be 0x800.
- IPv6 rules are supported on the Layer23 port. (But IPv6 destination MAC address in Layer23 mode is not supported.)

Flow action

OpenFlow actions does not change for Layer23 support. All actions currently supporting Layer 2 or Layer 3 flows continue to be supported. Actions currently supported are listed separately for different devices.

On RUCKUS ICX devices

When a matching flow entry is found, a set of actions can be applied for processing the packet. The system supports the following actions:

- Forward a packet to a port.
- Forward a packet to a set of ports.
- Forward a packet to a controller.

- Forward a packet received from a controller to a port or set of ports.
- Drop the packet.
- Keep, add, modify, or remove the VLAN ID or the VLAN priority. Modifying the VLAN ID per port is also supported (each destination port can send a packet with a different VLAN ID for the same matching rule).
- Modify the destination MAC address for Layer 2 flows.

Connecting to an OpenFlow Controller

To connect to an OpenFlow Controller in active mode, enter the following command:

```
device(config)# openflow controller ip-address 10.2.3.4
```

the IP address is the address of the OpenFlow Controller and SSL encryption is used by default. Also, the OpenFlow connection uses TCP port 6633.

To connect to an OpenFlow Controller in the passive mode, enter the following command:

```
device(config)# openflow controller passive no-ssl
```

You can optionally specify the TCP port to be used for the connection. By default, the device accepts the connection from a controller with any IP address. However, you can provide an IP address to limit which controller can connect to the device.

Use the **no** form of the command to remove a passive connection. Passive mode connections are intended for testing environments and not recommended for production environments.

Setting up SSL encryption for controller connections

By default, a connection to the controller uses SSL encryption. To set up SSL encryption, copy the SSL certificate and SSL client private key from the remote machine where you generated them into the device's flash using the following commands:

```
device(config)# copy tftp flash <remote ip> <remote file> client-certificate
device(config)# copy tftp flash <remote ip> <remote file> client-private-key
```

The IP address specifies the remote machine from which the SSL client certificate is being copied. The file name specifies the client certificate in the **copy tftp flash client-certificate** command, and the client private key in the **copy tftp flast client-private-key** command.

> **NOTE**
> SSL is not supported on passive controller connections.

The **remote file** variable specifies the file name of the client certificate in the first command, and the client private key in the second command.

For each controller, you must enter both the commands. The device can store up to three SSL certificates and client private keys. If you remove a controller connection, you must delete the SSL certificates and client private keys from the device's flash memory using the monitor mode commands.

> **NOTE**
> When using SSL to connect to the switch, the OpenFlow Controller can send only 50 flows at a time (for typical flows). This is not applicable to TCP-only connections to the OpenFlow controller.

Disabling an SSL client

You can disable the SSL client within the device using the following command:

```
device# ip ssl client disable
```

When you disable an SSL client in the device, the corresponding controller connection that used SSL encryption fails. However, you can re-enable the controller connection by removing the SSL encryption option from the controller connection. Use the **openflow controller ip-address no-ssl** command to disable SSL encryption in the connection.

Use the **no ip ssl client disable** command to re-enable the SSL client in the device.

Configuring multiple controller connections

Up to three controller connections are supported. You can configure these connections with active or passive modes, in any combination, such as all active, all passive, or some active and some passive. Each connection requires its own separate command. You can remove any of the connections using the **no** form of the **openflow controller ip-address** command. The following example shows how to configure three connections.

```
device(config)# openflow controller ip-address 10.2.3.4 no-ssl port 6635
device(config)# openflow controller ip-address 10.2.3.5 no-ssl
device(config)# openflow controller passive no-ssl ip-address 10.2.3.6
```

Configuring the system parameters for OpenFlow

You can specify the limit for OpenFlow flow table entries in the flow table using the following command:

```
device(config)# system-max openflow-flow-entries 304
```

You can specify the maximum number of flow table entries. The example shows 304 for the limit. The range is from 0 through 12000. The default is 1024 flow table entries.

Setting the system maximum

The maximum number of flows supported per device in a stack is 3000 in Layer 2 and Layer 3 modes and 1500 in the case of Layer23 mode or Layer 3 mode (with IPv6 matching).

The **system-max openflow-pvlan-entries** command sets the CAM size of OpenFlow protected VLAN entries for the device. By default, this value is set to 40.

```
device(config)# system-max openflow-pvlan-entries 30
```

The *value* variable represents the number of port and protected VLAN combination entries that can be configured in the system. The range is from 0 through 256.The above example shows 200 for the *value*. After using this command, you must reload the system.

The **system-max openflow-unprotectedvlan-entries** command sets the CAM size of OpenFlow unprotected VLAN entries for the device. By default, this value is set to 128.

```
device(config)# system-max openflow-unprotectedvlan-entries 100
```

The *value* variable represents the number of port and unprotected VLAN combination entries that can be configured in the system. The range is from 0 through 256. After using this command, you must reload the system.

Configuring the default action

By default, the device drops packets that do not match any of the programmed flows. However, you can configure a device-level option to forward the packets to the controller instead of dropping them. This is an optional configuration. If this option is not configured, packets that do not match any flow entries on a port are dropped. When sending a packet to the controller, a copy of the packet is sent to each of the configured controller connections.

To enable the default action, enter the following command:

```
device(config)# openflow default send-to-controller
```

Packets that match a flow entry on a port are processed according to the action specified and are not affected by this setting. Use the **no** form of the command to set the default action to drop such packets instead.

Displaying the OpenFlow status on the device

After enabling or disabling OpenFlow on a device, you can verify the configuration using any of the **show** commands.

Displaying the configured connections to controllers

Use the **show openflow** command to display the OpenFlow configuration, including the configured connections to controllers on the device. The output includes the configured unprotected VLANs as well.

```
device(config)# show openflow
Administrative Status:       Enabled
SSL Status:                  Enabled
Controller Type:             OFV 100
Number of Controllers:       1
Controller 1:
Connection Mode:       passive, TCP,
Listening Address:     0.0.0.0
Connection Port:       6633
Connection Status:      TCP_LISTENING
Match Capability:
L2 : Port, Source MAC, Destination MAC, Ether type, Vlan, Vlan PCP
L3 : Port, Vlan, Vlan PCP, Ethertype(IP,ARP,LLDP), Source IP, Destination IP, IP Protocol, IP TOS, IP Src
Port, IP Dst Port
L23: All
Normal Openflow Enabled Ports:
Openflow Hybrid Interfaces:
e1/1/1
Protected VLANs   : None
Unprotected VLANs :     2, 3, 4, 5, 6, 7, 8, 9, 10, 11
..................................................
..................................................
3994, 3995, 3996, 3997, 3998, 3999, 4000, 4001, 4011,
e1/2/1
Protected VLANs   : None
Unprotected VLANs :  4010,
Default action: drop
Maximum number of flows allowed: 65536
Active flow: 0
Maximum number of Protected Vlans allowed: 2048
Maximum number of Unprotected Vlans allowed: 4096
Total number of Unprotected Vlans: 4002
```

TABLE 5 Output fields for the show openflow command

Field	Description
Administrative Status	Indicates the administrative status of OpenFlow on the device.
Controller Type	Indicates the OpenFlow protocol version that is supported on the device.
Number of Controllers	Lists the number of controller connections configured on the device. Up to three concurrent controller connections are supported.

TABLE 5 Output fields for the show openflow command (continued)

Field	Description
Connection Mode	Indicates the mode of the controller connection configured. You can configure active or passive connection to controllers. An active connection is initiated by the device. In a passive connection, the device is in the listening mode, and accepts requests from controllers. If the optional controller address is not specified, any controller can establish a connection with the device in the passive mode. Refer to Connecting to an OpenFlow Controller on page 19.
Listening Address	Indicates the address of the specified controller.
Connection Port	Indicates the TCP port that is used for connection to the controller. By default, port 6633 is used.
Match Capability	Specifies the matching rules supported.
Normal OpenFlow Enabled Ports	Lists the ports on the device that are enabled for OpenFlow.
OpenFlow Hybrid Interfaces	Indicates the VLAN IDs.
Default action	Indicates the default action for packets that do not match any configured flows. By default, such packets are dropped. However, you can configure these packets to be sent to the controller by using the **openflow default send-to-controller** command.
Number of flows allowed	Indicates the maximum number of flows allowed on the device that is configured by using the **system-max openflow-flow-entries** command.

Displaying the data path ID of the device

OpenFlow associates a globally unique data path ID to be used by the controller to distinguish OpenFlow devices on a network. To display the data path ID assigned to the device, enter the following command:

```
device(config)# show openflow datapath-id

datapath-id# 0000001bedb3d0c0
```

The output of the command shows the data path ID. The data path ID is derived from the chassis MAC address.

Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the device and their statistics by using the following command:

```
device(config)# show openflow flows eth 1/1/2
```

The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed.

```
device(config)# show openflow flows
Flow ID: 1 Priority: 1 Status: Active
        Rule:
           In Port:      e1/1/1
           In Vlan:      Tagged[100]
            Vlan Mask:        0xfff
           Vlan PCP:     3
           Source Mac:   0000.0000.0001
           Destination Mac:      0000.0000.0002
           Source Mac Mask:      ffff.ffff.ffff
           Destination Mac Mask:         ffff.ffff.ffff
           Ether type:   0x00000800
           Source IP:            1.1.1.0        Subnet IP:       255.255.255.0
           Destination IP:       2.2.2.0        Subnet IP:       255.255.255.0
           IP TOS:       8
           IP Protocol:          17
           IP Protocol Source Port:      10000
```

```
           IP Protocol Destination Port:         80
            Cookie:        abcdef
            Cookie Mask:    0xffffff

      Timing Info:
           Idle Timeout                          :   500 secs
           Hard Timeout                          :  3000 secs
           Time Elasped(Since Flow Added)        :     6 secs
           Time Elasped(Since Last Packet Hit)   :     6 secs

        Instruction: Apply Action
        Action: FORWARD
           Out Port:  e1/1/2, Tagged, Vlan: 10
        Action: FORWARD
            Out Port: e1/1/3, Tagged, Vlan: 20
        Statistics:
            Total Packets: 0
       Total Bytes: 0
Flow ID: 10 Priority: 1 Status: Active
        Rule:
           In Port:      e1/1/17
           Ether type:   0x800
           Destination IP:       177.1.1.0      Subnet IP:       255.255.255.0
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port: normal
        Statistics:
           Total Pkts: 0
           Total Bytes: 0

Flow ID: 11 Priority: 1 Status: Active
        Rule:
           In Port:      e1/1/17
           Ether type:   0x800
           Destination IP:       180.1.1.0      Subnet IP:       255.255.255.0
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2
                        Out Port: normal
        Statistics:
           Total Pkts: 0
           Total Bytes: 0

Flow ID: 12 Priority: 1 Status: Active
        Rule:
           In Port:      e1/1/17
           Ether type:   0x800
           Destination IP:       188.1.1.0      Subnet IP:       255.255.255.0
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port: FLOOD
        Statistics:
           Total Pkts: 0
           Total Bytes: 0

Flow ID: 13 Priority: 1 Status: Active
        Rule:
           In Port:      e1/1/17
           Ether type:   0x800
           Destination IP:       199.1.1.0      Subnet IP:       255.255.255.0
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port: ALL
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

TABLE 6 Output fields for the show openflow flows command

Field	Description
Port	Port ID
VLAN	VLAN ID
Flow ID	An identifier for each flow. You can use the flow ID from this output to display flow-specific details.
Priority	The priority of the flow set by the controller when the flow is added, in the range 0 through 32768. If the priority value was not specified, the device assigns the default value, 32768. Priority 32768 has the highest priority. Priority 0 is reserved for unprotected VLAN support.
Status	Indicates whether the flow is configured correctly in the device. An active status indicates a correctly configured flow.
Rule	Here, the destination MAC Address Mask of FFFF.FFFF.FFFF indicates that only packets exactly matching the specified destination MAC address are forwarded.
Statistics	Indicates the counter of packets and bytes.

Setting the OpenFlow purge timer

You can configure the maximum time before stale flows are purged from the OpenFlow flow table after a switchover, failover, or operating system upgrade.

The valid range is from 1 through 600. The default is 240 seconds.

You may not need to change the value of the OpenFlow purge timer under normal circumstances. If you anticipate a delay in learning the flows from the controller after switchover, you can configure a larger value for the OpenFlow purge timer.

The following example shows how to set the OpenFlow purge timer:

```
device(config)# openflow purge-time 500
device(config)# no openflow purge-time 350
```

The **no** form of this command sets the purge timer time to its default value.

Administrating OpenFlow

Clearing the OpenFlow statistics

You can clear the flow statistics for all flows or, optionally, for a specified flow. Only the counters of packets and bytes (when applicable) are cleared, none of the other flow table entries are affected.

To clear flow counters, enter the following command:

```
device(config)# clear statistics openflow
```

To clear flow counters for a specified flow, enter the following command:

```
device(config)# clear statistics openflow 2
```

In the example, the flow counters are cleared only for flow ID 2. Use the **show openflow flows** command to obtain flow IDs.

Deleting the OpenFlow flows

You can delete an individual OpenFlow rule or all the flows in the flow table. To delete a single OpenFlow rule based on a flow ID, enter the following command:

```
device# clear openflow flowid 6
device# clear openflow flowid all
```

The **clear openflow** command deletes the rule irrespective of the state it is in (ACTIVE, PENDING_ADD, PENDING_MODIFY, or PENDING_DELETE). The same rule can be added again later from the controller if needed.

OpenFlow configuration considerations

After you enable OpenFlow on a device, you can configure, generate, and monitor flows on the ports configured on the device from a controller on OpenFlow-enabled ports. The RUCKUS ICX device flow table is entirely under the control of the OpenFlow Controller.

The OpenFlow Controller supports Administratively down (OFPPC_PORT_DOWN) through a Port Modification Message.

Behavior of ports and devices

- Ports that are enabled for OpenFlow cannot take part in any of the normal operations of the device, such as routing and Layer 2 forwarding. However, after OpenFlow is disabled on a port, the port can resume normal operations. This does not require disabling OpenFlow globally on the device.
- The flow table content is not cleared when the connection to a controller is lost. The device continues to forward traffic according to the flow entries defined in the flow table even in the absence of a controller connection.
- The flow table entries within the device are cleared when the device is reset.
- Flow table entries associated with a port are maintained when a port goes down. When the port comes back up, those flow entries are restored on the port. Flow entries are removed only with an explicit command from the controller.
- When OpenFlow is disabled globally on the device using the **no openflow enable** command, the flow table in the device is cleared. However, before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- When a controller tries to add a flow to the device with the same priority, rule, and action as a flow that exists in the flow table, the flow statistics are cleared (the system does not add a new flow). The following table summarizes the behavior for similar flows being successively added.

TABLE 7 Flow table behavior when flows similar to existing ones are added

Priority	Rule	Action	Device behavior
Same	Same	Same	Clear flow statistics
Same	Same	Different	- Update the action list - Clear the statistics
Same	Different	Same	Create new flow
Same	Different	Different	Create new flow
Different	Same	Same	Create new flow
Different	Same	Different	Create new flow

Removing an OpenFlow configuration from a device

In general, to remove OpenFlow from the device and make it a non-OpenFlow device, complete the following steps.

1. Disable OpenFlow on the ports where it is enabled.
2. Disable OpenFlow on the device globally.
3. (Optional) Set the maximum number of flows to zero using the **system-max openflow-flow-entries 0** command.
4. Reload the device.

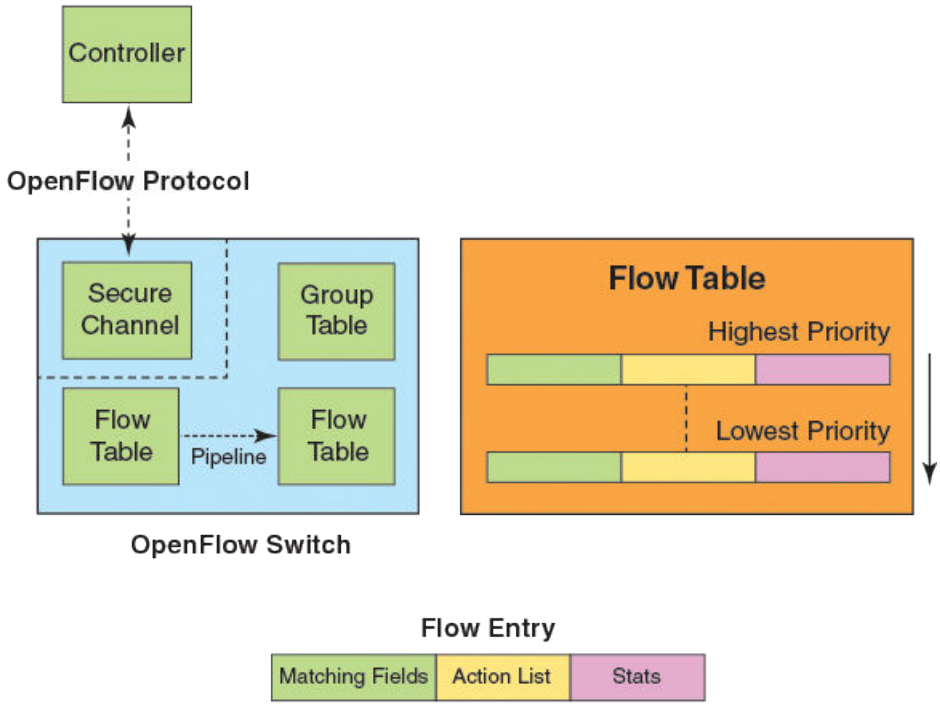
OpenFlow v1.3.0

Overview of OpenFlow v1.3.0

An OpenFlow switch maintains one or more flow tables, which are used for packet processing. The switch performs the actions listed in the table entry corresponding to the matched flow.

The OpenFlow Controller manages the OpenFlow switch using the OpenFlow. The OpenFlow Controller can add, delete, or modify flows by getting statistics for ports and flows and other information using the OpenFlow Protocol.

FIGURE 4 OpenFlow v1.3.0 architecture

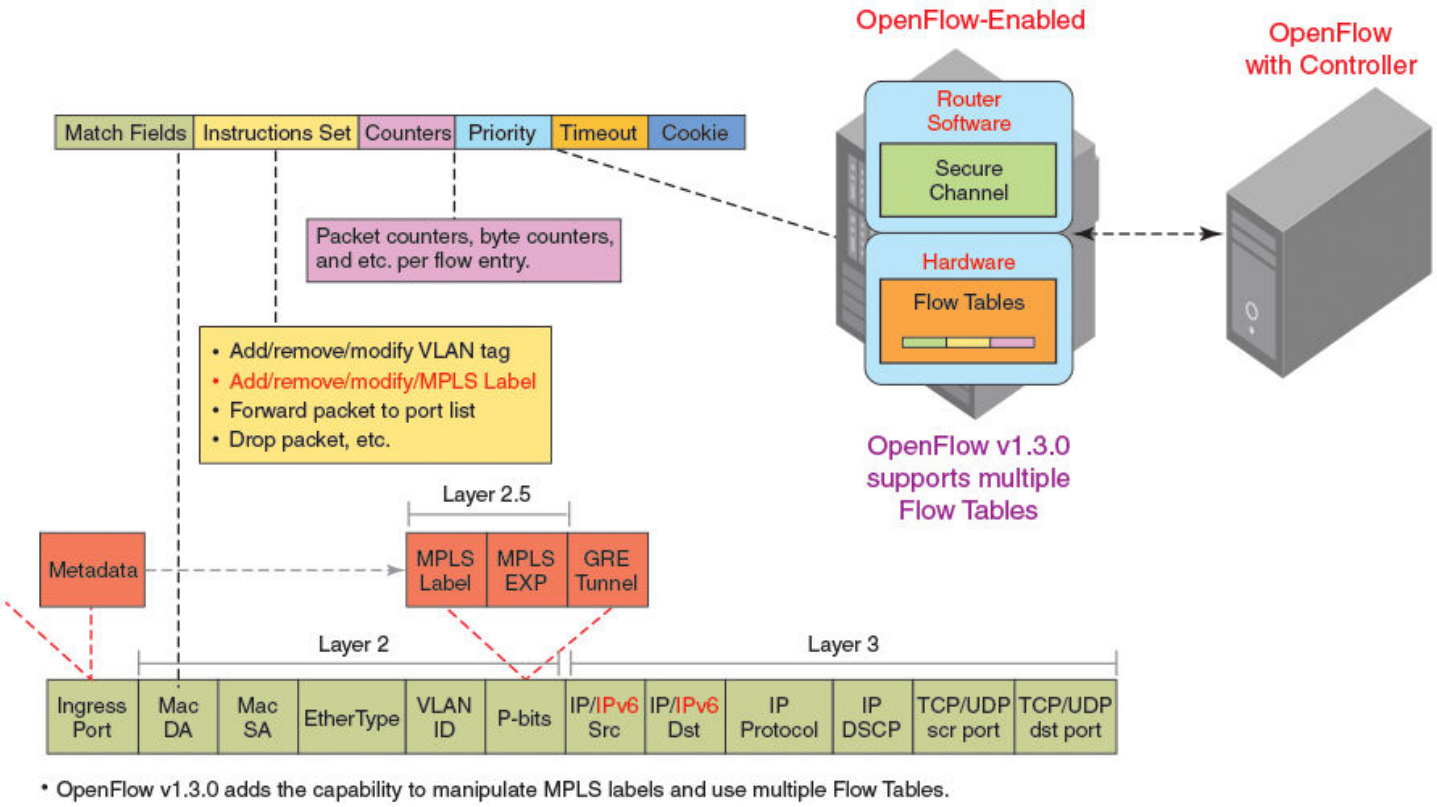


Each flow table maintained in a switch, consists of flow entries sorted by the flow priority. The highest priority flows are at the top of the flow table. Incoming packets are matched against the flow entries starting from the highest priority flow. If there is a match, then flow matching stops, and the set of actions for that flow entry are performed. The packets that do not match any flow entry, are either dropped or sent to the controller.

OpenFlow v1.3.0 defines three types of tables:

- Flow tables
- Group table
- Meter table

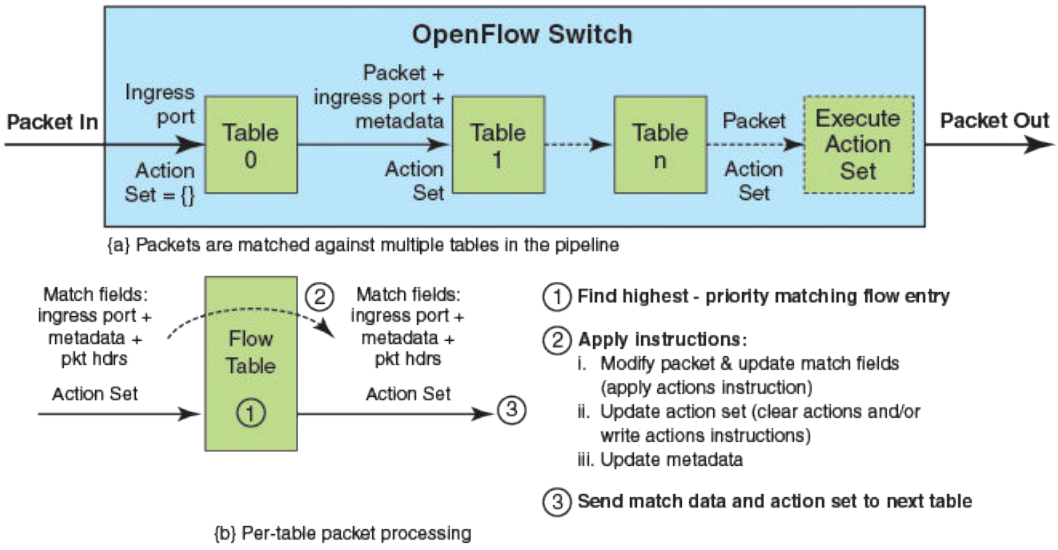
FIGURE 5 OpenFlow 1.3.0 flow table entries



The incoming packets are matched against the multiple tables in the pipeline.

NOTE
The ICX switches do not support matching or manipulating MPLS label information in a flow. Flows with MPLS ether type is considered as a failure.

FIGURE 6 Pipeline processing



NOTE

The ICX switches do not support multiple table lookup.

Flow table entries

Each flow table entry contains the fields described in the following table.

TABLE 8 Flow Table Entries

Field	Description
Match fields	The match fields consist of ingress ports, packet header fields, and metadata from a previous flow table
Priority	Matching precedence of the entry
Counters	Statistics for matching packets
Instructions	Action set or pipeline processing
Cookie	Opaque data sent by the OpenFlow Controller

RUCKUS ICX devices support the OpenFlow match fields in the following tables.

TABLE 9 OpenFlow match fields for ICX 7450

Match field	ICX 7450			Prerequisite	Description
	L2	L3	L23		
OXM_OF_IN_PORT	Yes	Yes	Yes	IN PORT present	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_IN_PHY _PORT	Yes	Yes	Yes	None	Physical port. In OFP_PACKET_IN messages, underlying physical port, when packet received on a logical port.
OXM_OF_ETH_DST	Yes	No	Yes	None	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	No	Yes	None	Ethernet source MAC address

TABLE 9 OpenFlow match fields for ICX 7450 (continued)

Match field	ICX 7450			Prerequisite	Description
	L2	L3	L23		
OXM_OF_Ether type	Yes	Yes	Yes	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes	Yes	None	VLAN-ID 802.1Q header
OFPVID_NONE	No	No	Yes	None	Match all untagged packets
OFPVID_PRESENT	No	No	No	None	Match packets with VLAN tag regardless of VLAN ID
OXM_OF_METADATA	No	No	No	None	Table metadata. Used to pass information between tables
OXM_OF_VLAN_PCP	Yes	Yes	Yes	None	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	No	Yes	Yes	Ether type= 0x0800	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	No	Yes	Yes	Ether type= 0x0800	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	No	Yes	No	Ether type= 0x0800	IPv4 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV4_DST	No	Yes	Yes	Ether type= 0x0800	IPv4 destination address. It can use subnet mask or arbitrary bit mask.
OXM_OF_TCP_SRC	No	Yes	Yes	IP PROTO = 6	TCP source port
OXM_OF_TCP_DST	No	Yes	Yes	IP PROTO = 6	TCP destination port
OXM_OF_UDP_SRC	No	Yes	Yes	IP PROTO = 17	UDP source port
OXM_OF_UDP_DST	No	Yes	Yes	IP PROTO = 17	UDP destination port
OXM_OF_SCTP_SRC	No	Yes	Yes	IP PROTO = 132	SCTP source port
OXM_OF_SCTP_DST	No	Yes	Yes	IP PROTO = 132	SCTP destination port
OXM_OF_ICMPV4 _TYPE	No	Yes	Yes	IP PROTO = 1	ICMP type
OXM_OF_ICMPV4 _CODE	No	Yes	Yes	IP PROTO = 1	ICMP code
OXM_OF_ARP_SPA	No	Yes	Yes	Ether type= 0x0806	IPv4 source address in the ARP payload. It can use subnet mask or arbitrary bit mask.
OXM_OF_ARP_TPA	No	Yes	Yes	Ether type= 0x0806	IPv4 destination address in the ARP payload.. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_SRC	No	Yes	No	Ether type= 0x86dd	IPv6 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_DST	No	Yes	Yes	Ether type= 0x86dd	IPv6 destination address. It can use subnet mask or arbitrary bit mask.

TABLE 10 OpenFlow Match Fields for ICX 7650

Match field	ICX 7650			Prerequisite	Description
	L2	L3	L23		
OXM_OF_IN_PORT	Yes	Yes	Yes	IN PORT present	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_IN_PHY _PORT	Yes	Yes	Yes	None	Physical port. In OFP_PACKET_IN messages, underlying physical port, when packet received on a logical port.
OXM_OF_ETH_DST	Yes	No	Yes	None	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	No	Yes	None	Ethernet source MAC address

TABLE 10 OpenFlow Match Fields for ICX 7650 (continued)

Match field	ICX 7650			Prerequisite	Description
	L2	L3	L23		
OXM_OF_Ether type	Yes	Yes	Yes	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes	Yes	None	VLAN-ID 802.1Q header
OFPVID_NONE	No	Yes	Yes	None	Match all untagged packets
OFPVID_PRESENT	No	No	No	None	Match packets with VLAN tag regardless of VLAN ID
OXM_OF_METADATA	No	No	No	None	Table metadata. Used to pass information between tables
OXM_OF_VLAN_PCP	Yes	Yes	Yes	None	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	No	Yes	Yes	Ether type= 0x0800	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	No	Yes	Yes	Ether type= 0x0800	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	No	Yes	No	Ether type= 0x0800	IPv4 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV4_DST	No	Yes	Yes	Ether type= 0x0800	IPv4 destination address. It can use subnet mask or arbitrary bit mask.
OXM_OF_TCP_SRC	No	Yes	Yes	IP PROTO = 6	TCP source port
OXM_OF_TCP_DST	No	Yes	Yes	IP PROTO = 6	TCP destination port
OXM_OF_UDP_SRC	No	Yes	Yes	IP PROTO = 17	UDP source port
OXM_OF_UDP_DST	No	Yes	Yes	IP PROTO = 17	UDP destination port
OXM_OF_SCTP_SRC	No	Yes	Yes	IP PROTO = 132	SCTP source port
OXM_OF_SCTP_DST	No	Yes	Yes	IP PROTO = 132	SCTP destination port
OXM_OF_ICMPV4 _TYPE	No	Yes	Yes	IP PROTO = 1	ICMP type
OXM_OF_ICMPV4 _CODE	No	Yes	Yes	IP PROTO = 1	ICMP code
OXM_OF_ARP_SPA	No	No	No	Ether type= 0x0806	IPv4 source address in the ARP payload. It can use subnet mask or arbitrary bit mask.
OXM_OF_ARP_TPA	No	No	No	Ether type= 0x0806	IPv4 destination address in the ARP payload.. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_SRC	No	Yes	No	Ether type= 0x86dd	IPv6 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_DST	No	Yes	Yes	Ether type= 0x86dd	IPv6 destination address. It can use subnet mask or arbitrary bit mask.

TABLE 11 OpenFlow Match Fields for ICX 7250

Match field	ICX 7250			Prerequisite	Description
	L2	L3	L23		
OXM_OF_IN_PORT	Yes	Yes	Yes	IN PORT present	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_IN_PHY _PORT	Yes	Yes	Yes	None	Physical port. In OFP_PACKET_IN messages, underlying physical port, when packet received on a logical port.
OXM_OF_ETH_DST	Yes	No	Yes	None	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	No	Yes	None	Ethernet source MAC address

TABLE 11 OpenFlow Match Fields for ICX 7250 (continued)

Match field	ICX 7250			Prerequisite	Description
	L2	L3	L23		
OXM_OF_Ether type	Yes	Yes	Yes	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes	Yes	None	VLAN-ID 802.1Q header
OFPVID_NONE	No	Yes	Yes	None	Match all untagged packets
OFPVID_PRESENT	No	No	No	None	Match packets with VLAN tag regardless of VLAN ID
OXM_OF_METADATA	No	No	No	None	Table metadata. Used to pass information between tables
OXM_OF_VLAN_PCP	Yes	Yes	Yes	None	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	No	Yes	Yes	Ether type= 0x0800	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	No	Yes	Yes	Ether type= 0x0800	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	No	Yes	No	Ether type= 0x0800	IPv4 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV4_DST	No	Yes	Yes	Ether type= 0x0800	IPv4 destination address. It can use subnet mask or arbitrary bit mask.
OXM_OF_TCP_SRC	No	Yes	Yes	IP PROTO = 6	TCP source port
OXM_OF_TCP_DST	No	Yes	Yes	IP PROTO = 6	TCP destination port
OXM_OF_UDP_SRC	No	Yes	Yes	IP PROTO = 17	UDP source port
OXM_OF_UDP_DST	No	Yes	Yes	IP PROTO = 17	UDP destination port
OXM_OF_SCTP_SRC	No	Yes	Yes	IP PROTO = 132	SCTP source port
OXM_OF_SCTP_DST	No	Yes	Yes	IP PROTO = 132	SCTP destination port
OXM_OF_ICMPV4 _TYPE	No	Yes	Yes	IP PROTO = 1	ICMP type
OXM_OF_ICMPV4 _CODE	No	Yes	Yes	IP PROTO = 1	ICMP code
OXM_OF_ARP_SPA	No	No	No	Ether type= 0x0806	IPv4 source address in the ARP payload. It can use subnet mask or arbitrary bit mask.
OXM_OF_ARP_TPA	No	No	No	Ether type= 0x0806	IPv4 destination address in the ARP payload.. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_SRC	No	Yes	No	Ether type= 0x86dd	IPv6 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV6_DST	No	Yes	Yes	Ether type= 0x86dd	IPv6 destination address. It can use subnet mask or arbitrary bit mask.

OpenFlow v1.3.0 instructions

Each flow entry has a set of instructions that are executed when the packet matches the entry.

The instruction set associated with each flow entry can have a maximum of one instruction of each type. The following table shows the actions supported on different RUCKUS ICX devices.

TABLE 12 Actions for flow table instruction

Actions	Description	Supported
Write-Action actions (Req)	Adds or overwrites specified actions to the action set.	Yes
Apply-Actions actions	Applies the specified actions immediately.	Yes

TABLE 12 Actions for flow table instruction (continued)

Actions	Description	Supported
Clear-Actions actions	Clears all the actions in the action set.	Yes
Meter meter-id	Directs the packet to the specified meter.	Yes
Goto -Table next-table-id (Req)	Indicates the next table in pipeline processing.	No
Write-Metadata metadata/mask	Writes the metadata field from the mask.	No
Output (Req)	Forwards the packet to a specified OpenFlow port. If out-port is Controller, then the packet is sent as packet-in message.	Yes
Drop (Req)	No explicit drop action. Packet with empty action set is dropped.	Yes
Group	Processes the packet through the specified group.	Yes
Set field	Modifies the values of the packet header based on the field type.	Yes
Push-Tag/ Pop-Tag	Adds and removes tag (newly inserted tags are always the outermost tags).	Yes
Set-Queue	Enqueues the packet to a specific queue on the outgoing port.	Yes
Decrement TTL	Decrements the TTL value by 1.	Yes
Normal Action mode	Processes the non-OpenFlow flows using local switching and routing.	Yes

The set fields in the following table are supported for OpenFlow instructions. The set field action is used to set the value in the header field.

TABLE 13 Supported set field action

Set field	Supported	Description
OXM_OF_ETH_DST	Yes	Ethernet destination MAC address (A maximum of 600 flows can be configured with this action)
OXM_OF_ETH_SRC	Yes	Ethernet source MAC address
OXM_OF_ETH_TYPE	No	Ether type of the OpenFlow packet payload after VLAN tags.
OXM_OF_VLAN_VID	Yes	VLAN-ID 802.1Q header (The output port must be a part of the VLAN that the flow is trying to set)
OXM_OF_VLAN_PCP	Yes	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	Yes	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_ECN	Yes	Modified ECN bits of the IP header.

OpenFlow v1.3.0 actions

Each flow has a set of instructions that are executed when the packet matches the flow as per OpenFlow v1.3.0 specifications. Each flow can have a maximum of one instruction of each type.

A switch can reject a flow entry, if it is unable to execute the instructions associated with the flow entry. In this case, the switch returns an unsupported flow error. Flow tables may not support every match, every instruction, or every action.

TABLE 14 Instructions for OpenFlow actions

Instruction	Description
actions	Adds specified actions to the action set.
next-table-id	Indicates the next table in pipeline processing (One table is supported).
meter-id	Directs the packet to the specified meter.
apply-actions	Applies the specified actions immediately. The packet is modified and subsequent matching in the pipeline is done on the modified packet.
clear-actions	Clears all the actions in the action set.

TABLE 14 Instructions for OpenFlow actions (continued)

Instruction	Description
write-metadata	Writes the metadata field from the mask.

RUCKUS ICX devices may support the actions listed in the following table.

TABLE 15 Supported OpenFlow actions

OpenFlow action	Supported
Process the packet through the specified group	Yes
Add and remove tag	Yes
Add newly inserted tags always as the outermost tags	Yes
No explicit drop action. Packet with empty action set should be dropped.	Yes
Modify the values of the packet header based on the field type	Yes
Modify the TTL value	No
Set the queue ID for the packet	Yes
Normal Action mode	Yes

Prerequisites for OpenFlow actions

The following prerequisites apply to the OpenFlow actions.

Decrement TTL

- Destination MAC address (DMAC) action is required, otherwise error is sent to controller.
- DMAC in the packet should be router MAC address of the device. This is not enforced as DMAC of the forwarding packet is unknown.
- Source MAC address (SMAC) is modified to router MAC address.
- VLAN header is stripped after the action.
- VLAN modification is supported (Push VLAN is not supported for tagged packets).

Multiple ports

- When action is to send the packets on multiple ports, then packet modification is not supported and an error message is generated, if there are any actions.

Send to controller action

- Send to controller action can be combined with single port or multiple port action.
- Send to controller action with single port does support packet modification.
- Send to controller action with multiple ports does not support packet modification.

VLAN modification

- The port on which the packet is to be forwarded with VLAN modification, should be part of the VLAN to be configured and the port must be added as tagged.

Destination MAC modification

- VLAN header is stripped after DMAC modification.
- VLAN modification is supported (Push VLAN is not supported for tagged packets).

Scaling considerations

These are the scaling considerations and limitations for the flows and CAM partitions.

- Each protected VLAN or unprotected VLAN requires 1 TCAM entry per interface.

Scaling numbers for flows

Only a few rules are used by the system to trap or to set QoS for control packets and security features. Up to 3,000 flows are supported. OpenFlow flows are categorized as into three types and are configured per interface.

- Layer 2: Only supports Layer 2 fields in match criteria; supports up to 512 flows.
- Layer 3: Only supports Layer 3 fields in match criteria; supports up to 1536 IPv4 flows and 768 IPv6 flows.
- Layer23: Supports Layer 2 and Layer 3 fields in match criteria; supports up to 512 flows.

Multiple controller connections

An OpenFlow switch may be connected to multiple controllers for reliability, allowing the switch to continue to operate in OpenFlow mode if a controller or controller connection fails. The controllers coordinate the management of the switch amongst themselves to help synchronize controller handoffs.

Each controller can have one of the following roles:

- Equal: The controller has full access to the switch. It receives all the asynchronous messages from the switch and sends commands to modify the state of the switch (add or delete flows).
- Slave: The controller has a read-only access to the switch. It does not receive the asynchronous messages (apart from port status). It does not execute commands that modify the state of the switch: **packet-out, flow-mod, group-mod, port-mod**, or **table-mod**. The switch must reply with an OFPT_ERROR message, if it receives one of those commands from a Slave controller. Other controller-to-switch messages are processed normally.
- Master: The controller has full access to the switch as in the Equal role. When the controller changes its role to Master, the switch changes the other controller in the Master role to have the Slave role. The role change does not affect controllers with the Equal role.

A switch can be simultaneously connected to multiple controllers in the Equal role, multiple controllers in the Slave role, and, at most, one controller in the Master role. Each controller can communicate its role to the switch by way of an OFPT_ROLE_REQUEST message. This message can be used by the controller to set and query the role of its channel with the switch.

To detect the out-of-order messages during a Master-to-Slave transition, the OFPT_ROLE_REQUEST message contains a 64-bit generation ID, filed by sequence number, that identifies the mastership view. The controllers coordinate the assignment of generation IDs. The generation ID is a monotonically increasing counter. A new (larger) value is assigned each time the mastership view changes; that is, when a new Master is designated. The generation ID value wraps around once the maximum value has been reached.

```
device(config)# openflow controller
------------------------------------------------------------------------
Contlr Mode  TCP/SSL IP-address     Port       Status            Role
------------------------------------------------------------------------
1   (Equal)    passive  TCP    0.0.0.0          6633   TCP_LISTENING
2   (Master)   active   TCP    10.25.128.179    6633   OPENFLOW_ESABLISHED
3   (Slave)    active   TCP    10.25.128.177    6633   OPENFLOW_ESABLISHED
3   (Equal)    active   TCP    10.25.128.165    6633   OPENFLOW_ESABLISHED
```

Asynchronous configuration

Asynchronous messages may need to be sent to multiple controllers. An asynchronous message is duplicated for each eligible OpenFlow channel, and each message is sent when the respective controller connection allows it.

A controller can also control which types of switch asynchronous messages are sent over its OpenFlow channel. This is done using an asynchronous configuration message that has the filter setting for all the messages.

Different controllers can receive different notifications. A controller in the Master role can selectively disable notifications, and a controller in the Slave role can enable notifications it wants to monitor.

Each controller configuration block for active connection maintains its own asynchronous configuration setting for every role. The default initial configuration is shown in the following table.

TABLE 16 Action for asynchronous configuration

Messages	Bit field	Master or Equal role	Slave role
Packet-in reasons	No_match	Enable	Disable
	Action	Enable	Disable
	Invalid_TTL	Enable	Disable
Port status reasons	Add	Enable	Enable
	Delete	Enable	Enable
	Modify	Enable	Enable
Flow removed reasons	Idle_timeout	Enable	Disable
	Hard_timeout	Enable	Disable
	Delete	Enable	Disable
	Group_delete	Enable	Disable

NOTE
The asynchronous messages Action and Invalid_TTL are not supported by RUCKUS ICX devices. Controllers can set these bits in the filter setting and the device can accept the bits, but the messages are not sent out by the device.

Supported OpenFlow messages

The following OpenFlow messages are supported.

TABLE 17 OpenFlow messages

Message type	Supported
OFPT_HELLO	Yes
OFPT_ERROR	Yes
OFPT_ECHO_REQUEST	Yes
OFPT_ECHO_REPLY	Yes
OFPT_EXPERIMENTER	No
OFPT_FEATURES_REQUEST	Yes
OFPT_FEATURES_REPLY	Yes
OFPT_GET_CONFIG_REQUEST	No
OFPT_GET_CONFIG_REPLY	No
OFPT_SET_CONFIG	No

TABLE 17 OpenFlow messages (continued)

Message type	Supported
OFPT_PACKET_IN	Yes
OFPT_FLOW_REMOVED	Yes
OFPT_PORT_STATUS	Yes
OFPT_PACKET_OUT	Yes
OFPT_FLOW_MOD	Yes
OFPT_GROUP_MOD	Yes
OFPT_PORT_MOD	No
OFPT_TABLE_MOD	No
OFPT_MULTIPART_REQUEST	Yes
OFPT_MULTIPART_REPLY	Yes
OFPT_BARRIER_REQUEST	Yes
OFPT_BARRIER_REPLY	Yes
OFPT_QUEUE_GET_CONFIG_REQUEST	No
OFPT_QUEUE_GET_CONFIG_REPLY	No
OFPT_ROLE_REQUEST	Yes
OFPT_ROLE_REPLY	Yes
OFPT_GET_ASYNC_REQUEST	Yes
OFPT_GET_ASYNC_REPLY	Yes
OFPT_SET_ASYNC	Yes
OFPT_METER_MOD	Yes

Hello-reply message

When a connection is initiated, the controller sends a hello message with a transaction ID. When the switch receives the hello message sent by the controller, it replies with another hello message using the same transaction ID as the received hello message. There are two hello messages sent from the switch to the controller during the connection establishment. Use the following command to disable the second hello message for unexpected interruption to the connection to the controller.

```
device(config) # no openflow hello-reply disable
```

The second hello message is enabled by default.

Packet-in message

When the default action or flow is send-to-controller, traffic is sent to the controller through the packet-in message. The VLAN ID for the packet is included in the OFPXMT_OFB_VLAN_VID match field of the packet-in message. Packet-in messages are handled as described below:

- For VLAN tagged traffic: The VLAN ID for the packet is included in the packet-in message.
- For untagged traffic: If the port is a member of the VLAN, the packet-in message includes their VLAN ID. If the port is not a member of any VLAN, no VLAN ID information is sent in the packet-in message.

Output port Normal action

Output port Normal is a reserved action. Normal action represents the traditional non-OpenFlow pipeline of the device. Normal is a special type of output port included in the actions associated with a flow. When a flow is received from the controller with an output port as Normal, the switch processes the matched incoming-packet using the local switching or routing.

Capabilities

Output ports with Normal action flows support the following capabilities:

1. Output port Normal action is supported on both hybrid ports and non-hybrid ports on the RUCKUSICX 7650, ICX 7450, and ICX 7250.
2. Generic flows with Normal action are supported .
3. Meter action is supported for Normal action flows.
4. Normal action is supported for both OpenFlow v1.0.0 and OpenFlow v1.3.0.
5. Normal action supports both tagged and untagged traffic.
6. Flows with Normal action can have additional actions as sent to the OpenFlow Controller.

Limitations

The following limitations apply to Normal action flows.

1. The following packet modifications are supported, when Normal is used as an output port in the flow:
   a. IP DSCP remark
   b. Set Queue
2. Group action is not supported for Normal action flows.
3. Normal action is not supported, if the OpenFlow port is an untagged port of a VLAN other than the default VLAN on the RUCKUSICX 7650, ICX 7450, and ICX 7250.

Outport port Normal and mirroring to a port

The port mirroring is applicable for monitoring real-time traffic. With OpenFlow, this application can monitor traffic on the selective flows for selective periods of time, inspect the packets in real time and take action.

1. A flow with action Normal and Output port is accepted and the packets are mirrored to Output port together with normal Layer 2 or Layer 3 processing.
2. This feature is supported only on the RUCKUSICX 7450 device.
3. None of the packet modifications is applied on the traffic, that is sent to the mirrored port.
4. All other capabilities and limitations listed in the output port Normal action are also applicable for port mirroring.

Output port All

A flow can have a reserved port All as output port instead of containing a list of the OpenFlow enabled ports. A flow with action set to output port All should be sent to all the OpenFlow ports.

- PACKET-OUT with action as All is sent to all OpenFlow ports.
- The port All supports both port based and generic flows.
- Packet modification is not supported with the action as output port All.

- OpenFlow disable on an interface is not allowed, if a flow with action All is present.

Output port Flood

Output action Flood indicates the packet to be flooded in the VLAN domain. This is similar to Normal action in case of the flow forwarding through hardware. For PACKET-OUT with action as Flood, VLAN information from the packet is extracted from PACKET-OUT and the packet is flooded on the ports that are part of that VLAN. In addition to VLAN, PACKET-OUT message contains IN-PORT, If a valid IN-PORT is present and the OpenFlow is enabled on that port then that port is excluded from the VLAN flood.

- When a VLAN is created, ports should be part of the VLAN.
- The hybrid and non-hybrid ports support Flood output ports.
- If VLAN is not created, then PACKET-OUT and message with Flood is dropped.
- The port Flood supports both port based and generic flows.

Supporting untagged traffic on OpenFlow hybrid ports on protected and unprotected VLANs

Untagged traffic is supported on protected VLANs or configured unprotected VLANs supporting IP traffic on an OpenFlow hybrid port. You can configure an untagged VLAN as a protected VLAN or an unprotected VLAN.

NOTE
To set the flow, the VLAN id is must if the port is untagged and have unprotected VLAN. Without the VLAN id, error is shown and the flow installation cannot be done.

NOTE
The openflow L2 or L3 lookup does not work on hybrid interfaces for default VLAN.

NOTE
On the RUCKUS ICX 7650, the following restrictions apply:

- Ports with OpenFlow hybrid mode enabled cannot be added to an untagged VLAN group.
- OpenFlow hybrid mode cannot be enabled on ports added to an untagged VLAN group.

OpenFlow port as an untagged member of only one VLAN

Assuming the port is configured as an OpenFlow hybrid port, the following cases are the configuration options to consider.

Case 1: When a port is added as untagged in an unprotected VLAN.

The configuration is accepted and untagged traffic on port 1/1/1, for example, is forwarded as per the OpenFlow rule if a matching rule is present. If a matching OpenFlow rule is not present, untagged traffic on port 1/1/1 is routed as per the routing table. If a route is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to the controller). DMAC should be the router MAC address to trigger Layer 3 routing as non-OpenFlow ports.

```
device(config-if-e10000-1/1/1)# openflow enable layer3 hybrid-mode
device(config-if-e10000-1/1/1)# vlan 300
device(config-vlan-300)# untagged ethernet 1/1/1
```

Case 2: When a port is added as untagged in a protected VLAN.

The configuration is accepted and untagged traffic on port 1/1/1, for example, is forwarded as per the route table.

```
device(config-if-e10000-1/1/1)# openflow enable layer3 hybrid-mode
device(config-if-e10000-1/1/1)# openflow protected-vlans 400
device(config-if-e10000-1/1/1)# vlan 400
device(config-vlan-400)# untagged ethernet 1/1/1
```

Case 3: When a port is removed as untagged from a configured unprotected VLAN.

The configuration is accepted and untagged traffic on port 2/1/1 is forwarded as per the OpenFlow rule if a matching rule is present. If a matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to the controller).

```
device(config-vlan-300)# no untagged ethernet 1/1/1
```

Case 4: When a port is removed as untagged from a protected VLAN.

The configuration is accepted and untagged traffic on port 1/1/1 is dropped.

```
device(config-vlan-400)# no untagged ethernet 1/1/1
```

Case 5: An untagged unprotected VLAN is configured as a protected VLAN on a port.

The configuration is accepted and untagged traffic on port 1/1/1 is forwarded as per the route table.

```
device(config-if-e10000-1/1/1)# openflow enable layer3 hybrid-mode
device(config-if-e10000-1/1/1)# vlan 300
device(config-vlan-300)# untagged ethernet 1/1/1
device(config-if-e10000-1/1/1)# openflow protected-vlans 300
```

Case 6: An untagged protected VLAN is removed from the port making it an untagged configured unprotected VLAN.

The configuration is accepted and untagged traffic on port 1/1/1, for example, is forwarded as per the OpenFlow rule if a matching rule is present. If a matching OpenFlow rule is not present, untagged traffic on port 1/1/1 is routed as per the routing table. If a route is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to the controller).

```
device(config-if-e10000-1/1/1)# openflow enable layer3 hybrid-mode
device(config-if-e10000-1/1/1)# openflow protected-vlans 400
device(config-if-e10000-1/1/1)# vlan 400
device(config-vlan-400)# untagged ethernet 1/1/1
device(config-if-e10000-1/1/1)# no openflow protected-vlans 400
```

Case 7: A configured untagged unprotected VLAN is deleted.

The configuration is accepted and untagged traffic on port 1/1/1 is forwarded as per the OpenFlow rule if a matching rule is present. If a matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to the controller).

```
device(config)# no vlan 300
```

Case 8: An untagged protected VLAN is deleted.

The configuration is accepted and untagged traffic on port 1/1/1 is dropped.

```
device(config)# no vlan 400
```

Idle and hard timeout support for OpenFlow

Each flow entry may have an idle timeout and or a hard timeout associated with it.

The idle timeout and a hard timeout control the removal of a flow entry from the OpenFlow table. If either value is non-zero, the switch must note the flow entry's arrival time, as it may need to evict the entry later. A non-zero idle_timeout entry field causes the flow entry to be removed after the given number of seconds, if no packet has been matched by the flow. A non-zero hard_timeout field causes the flow entry to be removed after the given number of seconds, regardless of how many packets it has matched.

Hard timeout: The absolute timeout after which the flow is removed from the device.

Idle timeout: The absolute timeout in which if there are no packets hitting the flow for the duration, then flow is removed from the device.

Idle timeout	Hard timeout	Behavior
Non-zero	Zero	Flow entry must expire after the idle timeout seconds with no received traffic.
Zero	Non-zero	Flow entry must expire in hard timeout seconds regardless of whether or not packets are hitting the entry.
Non-zero	Non-zero	Flow entry will timeout after idle timeout seconds with no traffic, or hard timeout seconds, whichever comes first.
Zero	Zero	Flow entry is considered permanent and it does not time out. It can be removed with a flow table modification message of type OFPFC_DELETE.

- The hard and idle timeout range is from 0 through 65535.
- Use the command **show openflow flows** to see the elapsed time between the establishment of the flow and the last time it was matched.
- The command also shows the active flows with idle and hard timeouts.
- When a flow gets deleted because of either idle or hard timeout, a syslog is generated. You can enable or disable message with the command**[no] openflow log timeout**.
- When timeout values are not specified in a flow, its value is displayed as zero and flow is considered as permanent till controller specifically deletes this flow.
- If OFPFF_SEND_FLOW_REM flag is set (in flow sent by the controller), then flow delete message is sent to the controller.
- In case of connection interruption, device continues to work under Fail Secure Mode.
- For modification requests (OFPFC_MODIFY or OFPFC_MODIFY_STRICT), if a matching entry exists in the table, the instructions field of this entry is updated with the value from the request, whereas its cookie, idle timeout, hard timeout, flags, counters, and duration fields are left unchanged.
- It is supported for both port based and generic flows.

Port-based flow

- When incoming traffic does not hit the flow, the idle time starts to increment. A hit causes this time to reset.
- The flow duration time keeps incrementing independent of the traffic hit.
- Once the limit is reached for either the idle time or the flow duration, the flow is deleted and the reason for deletion (idle timeout, hard timeout) is recorded by the system.

Generic flow

- Generic flows may be installed on multiple ports, depending on the OpenFlow-enabled ports. Even if the traffic hits only on one of those in_ports, flow statistics increments. Therefore, the idle time does not increment from its zero value.
- When incoming traffic does not hit any of the flows, the idle time starts to increment. A hit causes this time to reset.
- The flow duration time keeps incrementing independent of the traffic hit.

- Once the limit is reached for either the idle time or the flow duration, the flow is deleted and the reason for deletion (idle timeout, hard timeout) is recorded by the system.

Limitations

The completion of the deletion of a flow in a system lags the timeout by a small amount, typically not to exceed a maximum of four minutes. The lag depends on the number of flows present in the system at that time.

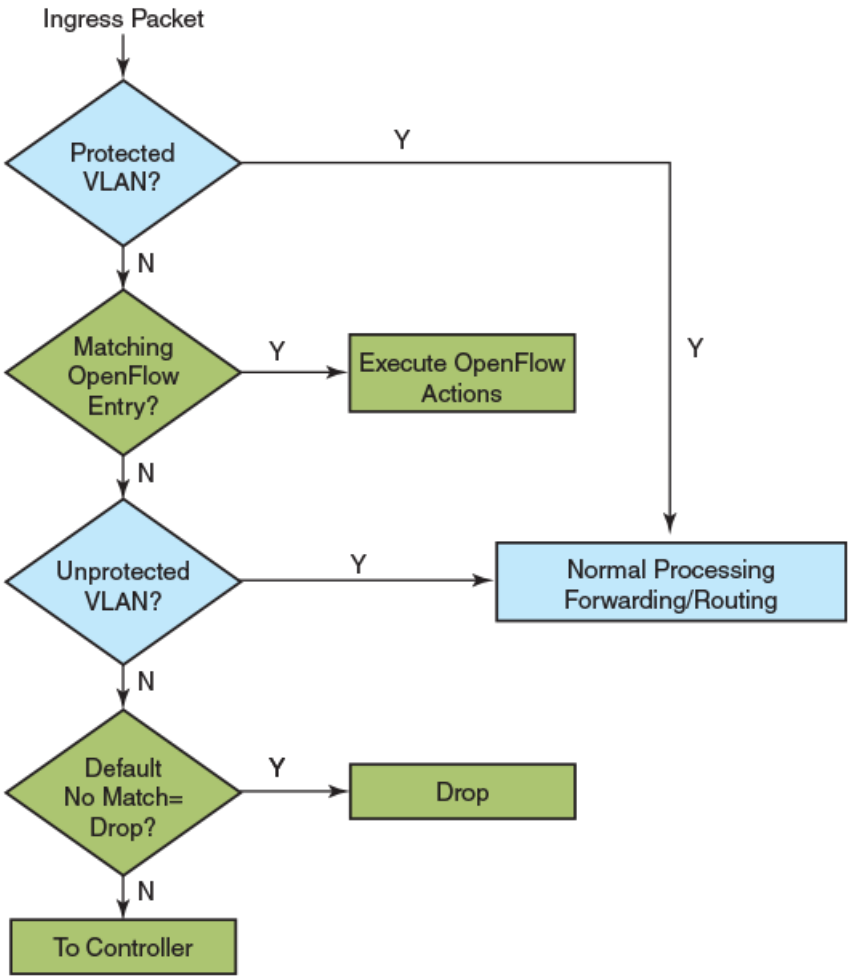
Layer 2 support for OpenFlow hybrid mode

The following Layer 2 features are supported in the OpenFlow hybrid mode, on protected VLANs and unprotected VLANs.

- Layer 2 switching and MAC learning
- STP
- LLDP, FDP, CDP
- LACP

The following diagram shows the flow for an ingress packet.

FIGURE 7 Packet flow diagram for Layer 2 support



Layer 2 switching and MAC learning

Source address MAC learning happens on the protected VLANs and on configured unprotected VLANs. The unconfigured VLAN traffic is dropped or sent to the controller based on the default rule.

On untagged VLANs, untagged traffic is flooded and the source is learned. The tagged traffic that matches the untagged VLAN is dropped. If the untagged VLAN becomes a protected VLAN, flow rules do not apply to untagged traffic. When the VLAN is configured as unprotected, the untagged traffic follows the matching flow rule in the presence of a flow.

When the tagged VLAN is a protected VLAN, flow rules do not apply on matching tagged traffic. If the VLAN is configured as an unprotected VLAN, the flow matching tagged traffic follows flow rule in the presence of flow. In the absence of flow, the default rule applies.

STP

The Spanning Tree Protocol (STP) can be enabled in the following ways:

1. When enabled globally, STP runs on all configured VLANs. STP runs on the OpenFlow hybrid ports with both tagged or untagged configured VLANs. The OpenFlow nonhybrid ports are not part of the STP instances.
2. On a per VLAN basis: When STP is enabled on a VLAN, all ports (OpenFlow hybrid ports) become part of the STP instance.
3. On a per-port basis: STP can be enabled on OpenFlow hybrid ports. STP is blocked on normal OpenFlow ports, so these ports are not part of any VLAN or STP instance.

In the absence of flows, STP works normally on OpenFlow hybrid ports. If the VLAN that is running STP becomes a protected VLAN, then OpenFlow flows are bypassed, but STP runs effectively. When the STP VLAN is configured as an unprotected and a matching OpenFlow rule is present, the OpenFlow rule overrides STP on both ingress and egress.

The STP Bridge Protocol Data Units (BPDUs) are tagged packets, which contain the VLAN ID on which the STP instance is running.

When you add a flow to match control packets, it may affect the convergence of Layer 2 protocols. For proper protocol convergence, the VLANs that are running STP on OpenFlow hybrid ports should be protected VLANs.

LLDP, FDP, CDP

The Link Layer Discovery Protocol (LLDP), Foundry Discovery Protocol (FDP) and Cisco Discovery Protocol (CDP) are supported on OpenFlow hybrid ports. The LLDP BPDUs are untagged packets. When untagged VLANs are protected, protocol convergence happens. When untagged VLANs are not protected and a matching OpenFlow rule is present, the PDUs are not processed. They take the OpenFlow path, and protocol convergence is affected.

LACP

The Link Aggregation Control Protocol (LACP) is supported on a keep-alive LAG (singleton link) on the OpenFlow hybrid port. The LACP PDUs are untagged packets. When untagged VLANs become protected, protocol convergence happens. When untagged VLANs are not protected and a matching OpenFlow rule is present, the PDUs are not processed. They take the OpenFlow path, and protocol convergence is affected.

Limitations

- Layer 2 hybrid mode is supported in Layer 2 and Layer 23 modes only.
- LACP support on keep-alive LAG is limited to OpenFlow hybrid ports only.
- Traffic is not forwarded on STP blocked ports, even when matching OpenFlow flow is present.
- Source MAC learning happens for an unprotected VLAN in the presence of matching OpenFlow flow as well.

Group table

The group table introduces the ability to add support for port group abstraction for multi-pathing. This enables OpenFlow to represent a set of ports as a single entity for forwarding packets.

The group table supports the following group types:

- All: Executes all the buckets in the group; mostly used for flooding and multicasting.
- Indirect: Executes one defined bucket in the group. The action taken by this group type is sending packets to the next hop.
- Select: Executes one bucket in the group. The action bucket is chosen by a switch-defined algorithm, such as round robin or hashing (for example, load sharing).

- Fast failover: Executes the first live bucket, used in cases such as redundancy.

A group table consists of group entries. The counters in the following table are available in a group entry.

TABLE 18 Group entry counters

Counter	Description
Group Identifier	A 32-bit unsigned integer uniquely identifying the group
Group type	Determines group semantics
Counter	Number of packets processed by a group
Action bucket	Ordered list of action buckets, where each action bucket contains a set of actions to execute and associated parameters

Scaling group numbers

The output of the **show openflow groups** command displays the maximum number of actions in a bucket, the maximum number of buckets in a group and the maximum number of groups for scaling the group in OpenFlow.

```
device(config)# show openflow groups 20
```

On RUCKUS ICX devices

- The maximum number of actions in a bucket is 1.
- The maximum number of buckets in a group is 64. The maximum number of action buckets for group Select is 32.
- The maximum number of groups is 512; for group Select and group All, the maximum number of groups are 512 also.

Considerations and limitations for group tables

You must take into account the following when you configure group tables for OpenFlow flows.

For configuring group tables

- RUCKUS ICX devices support all group types in the OpenFlow v1.3.0 specification.
- Multiple actions are supported by a group. The following packet modifications are supported through these actions for group types Indirect, Select and Fast failover.
    1. Modify a VLAN
    2. Modify the source MAC address
    3. Modify the destination MAC address
    4. Decrement the TTL
    5. Pop VLAN
- Each action bucket can have only one output port.
- Each OpenFlow port can be a part of any number of groups.
- A group entry can include ports from different slots and ports with different speeds.
- Group tables are not impacted based on the OpenFlow type on the interface (Layer 2 or Layer 3 or Layer23 and hybrid interfaces).
- To disable OpenFlow on interfaces, the interface must be removed from any group entry first.

Limitations

For configuring OpenFlow, consider the following limitations.

- Watch_group is not supported in Fast failover group type.
- PBR or Transparent VLAN flooding cannot be configured along with the group table, when OpenFlow v1.3.0 is enabled and vice versa.

The following additional limitations apply to the specific group types.

For group All

To multicast flow-matching traffic to all action buckets, all action buckets are executed every time for the group All.

- A packet is replicated for the output port in each bucket. Only one packet is processed for each bucket of the group.

For group Indirect

The group Indirect executes one defined action bucket in a group. Only one action bucket can exist and it is executed every time.

Group Indirect supports one and only one bucket in each group entry.

For group Select

To load balance flow-matching traffic to all action buckets, one of the action buckets is chosen each time for the group Select.

- Weighted load balancing for group Select is not supported.
- Group chaining is not supported.
- Individual bucket statistics are not supported.

For group Fast failover

The group Fast failover executes the first live bucket. Each action bucket associated with a specific port or group determines the liveness of the bucket.

- The buckets are selected in the defined sequence.
- On a stack unit Fast failover, traffic convergence takes up to 3 seconds.
- If no buckets are live, packets are dropped.

Group events

The following group events are supported by OpenFlow.

- Add group
- Delete group
- Add port to the group
- Delete port from the group
- Group type modification
- Group output port is up
- Group output port is down

Statistics

Group statistics are cumulative flow statistics that use the group ID in the action list. The following statistics are supported per group:

- Reference count (flow entries)
- Packet count (limited support on different devices)
- Byte count

For OpenFlow hybrid ports

- A group table does not affect hybrid functionality.
- Flows within a group on the hybrid port are treated the same as other flows.
- A group can support Normal and the hybrid OpenFlow port together.

Enqueue

The controller is able to set up and configure queues and then map flows to a specific queue. The queue configuration sets the queue ID for a packet and determines the queue to be used for scheduling and forwarding the packet.

Queue configuration takes place outside the OpenFlow protocol based on weights for a particular queue using Weighted Round Robin (WRR) scheduling.

There are two distinct parts that form the enqueue mechanism:

- Configuration
- Flow-queue mapping or forwarding

Assuming that a queue is already configured, you can associate a flow with an OFPAT_ENQUEUE action which forwards the packet through the specific queue on a port. Note that an enqueue action overrides any TOS or VLAN_PCP-related behavior that is potentially defined in the flow, but the packet is not changed or modified due to an enqueue. A total of 8 queues per port are supported.

In case of stacking, queue 7 is reserved for stacking messages. Any queue set to 7 is reclassified to queue 6. When there is no stacking, the standalone queue set to 7 remains as 7.

Use case: OpenFlow meter and enqueue

QoS is usually implemented to provide appropriate levels of service to support Service Level Agreements (SLAs). You have the ability to meter and determine customer traffic according to the bandwidth guaranteed provided to the customer by way of a combination of OpenFlow v1.0.0 or v1.3.0 actions. The policing must be fine grained and flexible enough as supported by OpenFlow match semantics. For instance, the match criteria for rate limiting one application may be based on a VLAN tag and, for another application, it may be based on the Layer 4 UDP or TCP port. The confirm action sets the appropriate queue ID for the packets, while the exceed action may cause the traffic to be dropped in case of congestion or remarked to a lower priority and with a different queue ID. When the packet is forwarded to a port using the output action, the queue ID determines which queue attached to this port is used for scheduling and forwarding the packet.

Configuring OpenFlow enqueue

Queue configuration takes place outside the OpenFlow protocol, either through a command line tool or through an external dedicated configuration protocol.

The minimum guaranteed bandwidth is configured through assignment of weights for a particular queue (with WRR scheduling).

Complete the following steps to configure OpenFlow enqueue.

1. Enable queue statistics at the global level.

```
device (config) # statistics
device (config-statistics # tm-voq-collection
```

2. Configure WRR scheduling and weights for the queues at the egress.

```
device(config-if-e10000-1/2/5)# qos scheduler weighted 10 10 20 10 10 20 10 10
```

3. Configure Shaper configuration for the queues at the egress port (configuring maximum rate).

```
device(config-if-e10000-1/2/5)# qos shaper priority 3 3000
```

4. Disable the encode policy map at the egress port.

```
device(config-if-e10000-1/2/5)# qos pcp encode-policy off
device(config-if-e10000-1/2/5)# qos dscp encode-policy off
```

5. Configure priority queues from 8 to 4 or vice versa.

```
device(config)# system-max-tm-queues 4
```

The queues are now configured for forwarding actions. After the queues have been configured, flows can be mapped to queues and packets are forwarded through them.

Limitations

The following limitations apply to the enqueue:

- A flow can have a maximum of one queue ID which is applicable for all output port in the action list.
- OpenFlow flows with action as Set IP TOS or Set VLAN PCP cannot be supported simultaneously with enqueue configuration. Such a configuration is rejected.
- QoS functionality of hybrid traffic flowing through these ports is affected.

Metering

Per-flow metering measures and controls the rate of packets for each flow entry. Per-flow meters enable OpenFlow to implement simple QoS operations, such as rate limiting, and can be combined with per-port queues to implement complex QoS frameworks, such as DiffServ.

Meters are attached directly to flow entries. Each meter can have one or more meter bands. Each meter band specifies the rate of the band and the way packets are processed (DROP or DIFFSERV). OpenFlow metering operation is similar to ingress rate limiting in a QoS operation.

A meter table consists of meter entries. The counters in the following table are available in the meter entry.

TABLE 19 Meter Entry

Counter	Description
Meter Identifier	A 32-bit unsigned integer uniquely identifying the meter
Meter band	A list of meter bands, where each meter band specifies the rate of the band and the way to process the packet. Rate and burst size are based on the line rate of the data traffic in contrast to the information rate.
Counter	Number of packets processed by a meter

Packets are processed by a single meter band based on the current measured meter rate. The meter applies the meter band with the highest configured rate that is lower than the current measured rate. If the current rate is lower than any specified meter band rate, no meter band is applied.

TABLE 20 Supported Meter Bands

Meter bands	Supported
DROP	Yes
DSCP_REMARK	No
EXPERIMENTER	No

Each band type contains the following meter configuration parameters from the controller:

- Rate value in kbps
- Rate value in packets per second
- Burst size
- Statistics collection

TABLE 21 Meter Configuration Parameters

Configuration flags	Supported
OFPMF_KBPS	Yes
OFPMF_PKTPS	No
OFPMF_BURST	Yes
OFPMF_STATS	Yes

The metering system supports the features in the following table.

TABLE 22 Metering Capabilities Supported for Metering Features

Feature	On RUCKUS ICX devices
Maximum meter available in the system	1024
Band types (bitmap)	DROP
Capabilities (bitmap)	KBPS, BURST, STATS
Maximum number of band per meter	1 or 2
Maximum color value	2

Meter statistics

The following statistics are supported per meter:

- Flow count (number of flows associated with the meter)
- Input byte count (cumulative byte count on all associated flows)
- Duration (second)
- Duration (nanosecond)(optional)

The flow and the byte count calculate all packets processed by the meter. The duration fields indicate the elapsed time for which the meter has been installed on the device.

The following counters are supported for meter band type:

- Band packet count
- Band byte count

The byte band count presents the total numbers for all bytes processed by the band.

TABLE 23 Meter band statistics

Band Type	Meter Band statistics	Supported
DROP	In band packet count	Yes
	In band byte count	Yes
DSCP_REMARK	In band packet count	No
	In band byte count	No

Limitations

The following limitations apply to the RUCKUS ICX devices for metering.

Meter band

The following limitations apply to the meter bands:

- The minimum burst size for the DSCP or DROP band is 82 kbps and the maximum is 17,179,600 kbps.
- The maximum rate for DROP or DSCP is 1,000,000 kbps; the minimum is 64 kbps.
- The DSCP band rate cannot be greater than the DROP band rate.
- The precedence level for the DSCP band type should always be 1.

The maximum number of meters for the devices is 1024.

Displaying OpenFlow meters

A meter measures the rate of packets assigned to it and enables controlling the rate of those packets.

The hardware resources are shared between OpenFlow and other features, so these resources are allocated on a first-come-first-serve basis.

Enter the **show openflow meters** command.

The following example output shows with a single meter band.

```
device(config)# show openflow meters 1
Meter id: 1

  Transaction id:        1437
  Meter Flags:            KBPS BURST STATS
  Flow Count:            0
  Number of  bands:      1
  In packet count:       -NA-
  In byte count:         0

  Band Type:    DROP

    Rate:                         750000
    Burst size:                   1500          kb
    In packet band count:         -NA-
    In byte band count:           0
```

The following example output shows information for 2 meter bands.

```
device(config)# show openflow meters 2
Meter id: 2

  Transaction id:        1438
  Meter Flags:            KBPS BURST STATS
  Flow Count:            0
  Number of  bands:      2
  In packet count:       -NA-
  In byte count:         0

  Band Type:    DSCP-REMARK

    Rate:                          750000
    Burst size:                    1500          kb
    Prec level:                    1
    In packet band count:          -NA-
    In byte band count:            0


  Band Type:    DROP

    Rate:                          1000000
    Burst size:                    2000          kb
    In packet band count:          -NA-
    In byte band count:            0
```

Meter implementation does not address any vendor-specific proprietary messages.

COMMSCOPE®
RUCKUS®

350 West Java Dr., Sunnyvale, CA 94089 USA
https://www.commscope.com