53-1003756-01
March 2015

Brocade FastIron OpenFlow

Deployment Guide

BROCADE

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

Contents

Brocade FastIron OpenFlow Deployment Guide

Preface

OpenFlow protocol provides a standard programmatic interface to program network switches enabling administrators to decouple the control plane from the forwarding plane. This makes network deployments easier to deploy and manage with centralized control. OpenFlow is currently available in 2 versions 1.0 and 1.3. The scope of this document is limited to the flows with regular matches and actions. Meters and groups will be covered in future versions of this document.

This document is prepared based on Brocade ICX6610 running Brocade FastIron software version 08.0.20

Purpose of this document

This guide discusses two use cases:

- Application-based resource allocation

  To guarantee the resources for traffic belonging to a specific application
- Role-based access control

  To restrict/allow access to the network based on the privileges assigned to a user by the administrator.

Audience

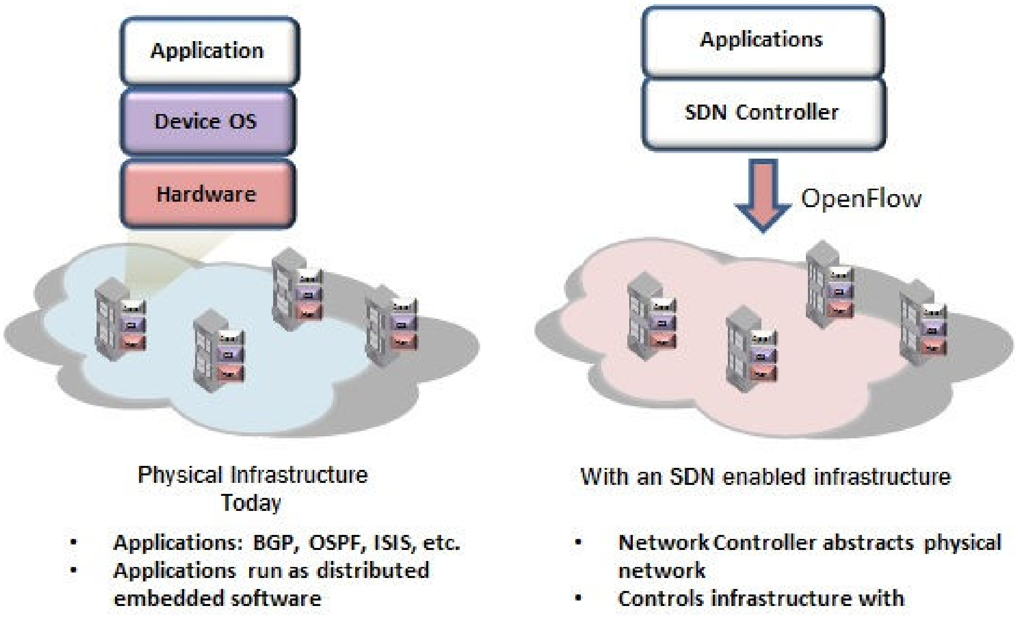
This document can be used by technical marketing engineers and system engineers to help customers understand the use case scenarios where OpenFlow can be deployed.

Introduction

Software Defined Network (SDN) is a software abstraction layer on top of networking infrastructure. SDN abstracts physical networking and makes networking hardware vendor independent. Some of the key benefits of SDN are:

- Decoupling of network application innovation from router OS release schedules.
- Accelerate automation of network changes to increase service velocity.

FIGURE 1 SDN overview


Application
Device OS
Hardware
Applications
SDN Controller
OpenFlow
Physical Infrastructure Today
• Applications: BGP, OSPF, ISIS, etc.
• Applications run as distributed embedded software
With an SDN enabled infrastructure
• Network Controller abstracts physical network
• Controls infrastructure with

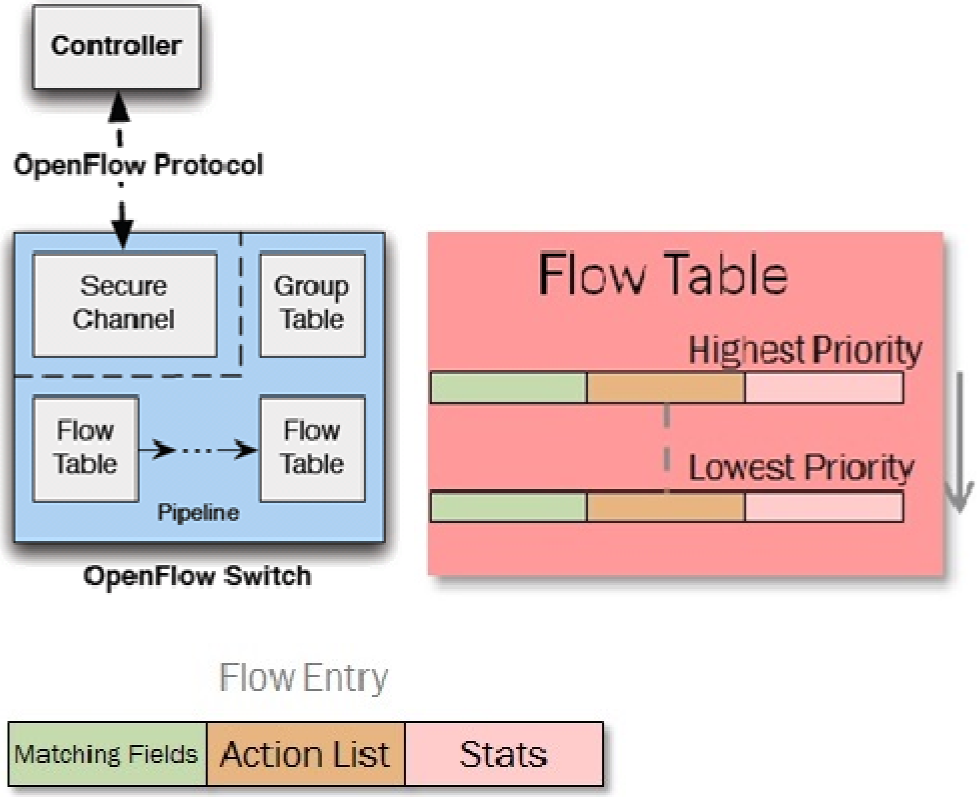
The OpenFlow specification is developed by Open Networking Foundation (ONF).

http://www.opennetworkingfoundation.org/

OpenFlow specifications started with v1.0.0. Currently v1.3.2 is available (v1.4.0 is still being evaluated). Most controllers and switches in the market today support only v1.0.0 and v1.3.0(2) is being rolled out gradually and is limited by the availability of Controllers. This document covers OpenFlow v1.3.0.

The figure shown below explains the OpenFlow 1.3 architecture. As shown, the OpenFlow switch connects to the controller over a TCP session (non-encrypted connection or secure SSL connection). Upon the connection establishment, the controller sends flows to the switch. The controller may also send group and meters. The OpenFlow switch installs the flows in the hardware (TCAM) based on the matching and actions mentioned in the flows.

FIGURE 2 OpenFlow 1.3 architecture



OpenFlow 1.3

OpenFlow 1.3 brings significant advantages over the 1.0 specification. The main additions include:

- Multiple flow tables vs. the single flow table in 1.0
- Support for IPv6 address matching
- Meter tables
- Group tables

Components of OpenFlow on the switch

OpenFlow 1.3 on the switch comprises of the following components:

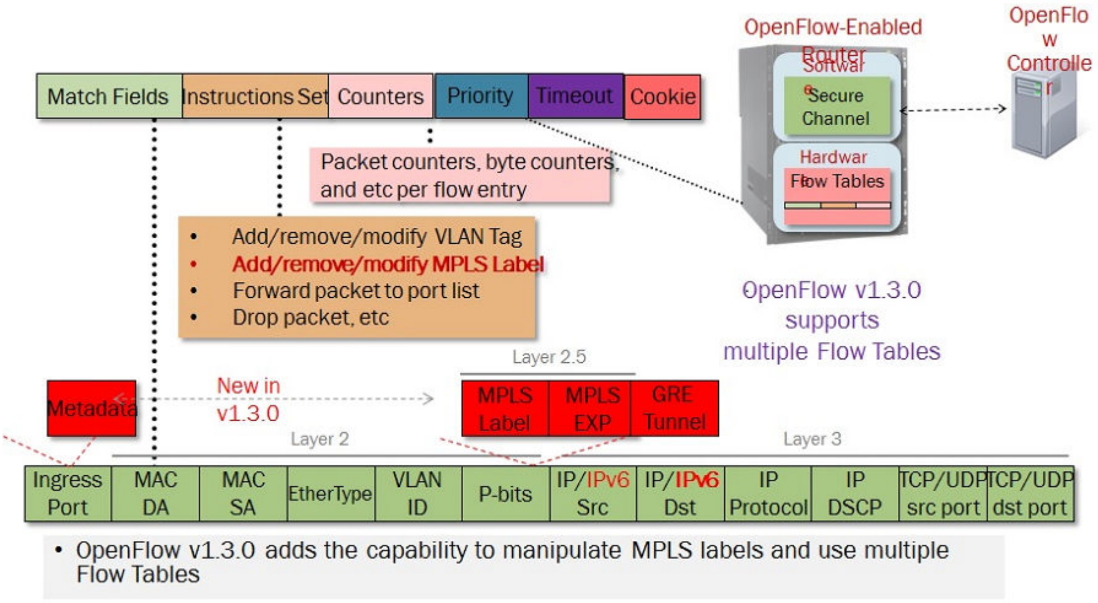
- Flow tables
- Meter tables
- Group tables

In the context of this document, the 'switch' refers to the Brocade ICX6610 switch running the firmware version 08.0.20.

Flow table

A flow table consists of flow entries sorted with the flow priority. Highest priority flows are at the top of the flow table. Incoming packets are matched against the flow entries in the order. If there is a match, then flow matching stops and the set of actions for that flow entry are performed. Packets that do not match any flow entry are either dropped or sent to the controller.

FIGURE 3 OpenFlow 1.3 flow



- OpenFlow v1.3.0 adds the capability to manipulate MPLS labels and use multiple Flow Tables

Flow Match Fields

The following table lists the match fields supported with any pre-requisites.

The match fields mentioned are defined in OpenFlow 1.3 Specification.

TABLE 1 Flow Match Fields

oxm_ofb_match-fields	ICX6610			Prerequisite
	L2-Mode	**L3-Mode**	**L23-Mode**	
OXM_OF_IN_PORT	Yes	Yes	Yes	None
OXM_OF_IN_PHY_PORT	Yes	Yes	Yes	IN PORT present
OXM_OF_VLAN_VID	Yes	Yes	Yes	None
OXM_OF_METADATA	No	No	No	None
OXM_OF_ETH_DST	Yes	No	Yes	None
OXM_OF_ETH_SRC	Yes	No	Yes	None
OXM_OF_ETH_TYPE	Yes	No	Yes	None
OXM_OF_VLAN_PCP	Yes	No	Yes	VLAN VID!=NONE
OXM_OF_IP_DSCP	No	No	No	ETH_TYPE=0x0800

TABLE 1 Flow Match Fields (Continued)

oxm_ofb_match-fields	ICX6610			Prerequisite
	L2-Mode	L3-Mode	L23-Mode	
OXM_OF_IP_ECN	No	No	No	ETH_TYPE=0x0800
OXM_OF_IP_PROTO	No	Yes	Yes	ETH_TYPE=0x0800
OXM_OF_IPV4_SRC	No	Yes	Yes	ETH_TYPE=0x0800
OXM_OF_IPV4_DST	No	Yes	Yes	ETH_TYPE=0x0800
OXM_OF_TCP_SRC	No	Yes	Yes	IP PROTO=6
OXM_OF_TCP_DST	No	Yes	Yes	IP PROTO=6
OXM_OF_UDP_SRC	No	Yes	Yes	IP PROTO=17
OXM_OF_UDP_DST	No	Yes	Yes	IP PROTO=17
OXM_OF_SCTP_SRC	No	Yes	Yes	IP PROTO=132
OXM_OF_SCTP_DST	No	Yes	Yes	IP PROTO=132
OXM_OF_ICMPV4_TYPE	No	Yes	Yes	IP PROTO=1
OXM_OF_ICMPV4_CODE	No	Yes	Yes	IP PROTO=1
OXM_OF_ARP_OP	No	No	No	ETH_TYPE=0x0806
OXM_OF_ARP_SPA	No	Yes	Yes	ETH_TYPE=0x0806
OXM_OF_ARP_TPA	No	Yes	Yes	ETH_TYPE=0x0806
OXM_OF_ARP_SHA	No	No	No	ETH_TYPE=0x0806
OXM_OF_ARP_THA	No	No	No	ETH_TYPE=0x0806
OXM_OF_IPV6_SRC	No	Yes	Yes	ETH_TYPE=0x86dd
OXM_OF_IPV6_DST	No	Yes	No	ETH_TYPE=0x86dd
OXM_OF_IPV6_FLABEL	No	No	No	ETH_TYPE=0x86dd
OXM_OF_ICMPV6_TYPE	No	No	No	IP PROTO=58
OXM_OF_ICMPV6_CODE	No	No	No	IP PROTO=58
OXM_OF_IPV6_ND_TARGET	No	No	No	ICMPV6 TYPE=135 or ICMPV6 TYPE=136
OXM_OF_IPV6_ND_SLL	No	No	No	ICMPV6 TYPE=135
OXM_OF_IPV6_TLL	No	No	No	ICMPV6 TYPE=136

OXM_OF_MPLS_LABEL	No	No	No	ETH_TYPE=0x8847
				or
				ETH_TYPE=0x8848

TABLE 1 Flow Match Fields (Continued)

oxm_ofb_match-fields	ICX6610			Prerequisite
	L2-Mode	L3-Mode	L23-Mode	

OXM_OF_MPLS_TC	No	No	No	ETH_TYPE=0x8847
				or
				ETH_TYPE=0x8848

OXM_OF_MPLS_BOS	No	No	No	ETH_TYPE=0x8847
				or
				ETH_TYPE=0x8848

OXM_OF_PBB_ISID	No	No	No	ETH_TYPE=88E7
OXM_OF_TUNNEL_ID	No	No	No	None
OXM_OF_IPV6_EXTHDR	No	No	No	ETH_TYPE=0x86dd

OpenFlow 1.3 Supported Instructions

TABLE 2 Flow Instructions

ofp_instruction_type	Yes/No	Description
OFPIT_GOTO_TABLE	No	Set up the next table in the lookup pipeline
OFPIT_WRITE_METADATA	No	Set up the metadata field for use later in pipeline
OFPIT_WRITE_ACTIONS	Yes	Write the action(s) onto the data path action set
OFPIT_APPLY_ACTIONS	Yes	Applies the action(s) immediately
OFPIT_CLEAR_ACTIONS	Yes	Clears all actions from the data path action set
OFPIT_METER	Yes	Apply meter rates (rate limiter)
OFPIT_EXPERIMENTER	No	Experimenter instruction

OpenFlow Supported Actions

TABLE 3 Flow Actions

ofp_action_type	ICX6610	Description
OFPAT_OUTPUT	Yes	Output port
OFPAT_COPY_TTL_OUT	No	Copy TTL "outwards" - from the next-to-outermost to outermost
OFPAT_COPY_TTL_IN	No	Copy TTL "inwards" - from the next-to-outermost to outermost
OFPAT_SET_MPLS_TTL	No	MPLS TTL
OFPAT_DEC_MPLS_TTL	No	Decrement MPLS TTL
OFPAT_PUSH_VLAN	Yes	Push a new VLAN tag - with restrictions

TABLE 3 Flow Actions (Continued)

ofp_action_type	ICX6610	Description
OFPAT_POP_VLAN	Yes	Pop the outer VLAN tag
OFPAT_SET_QUEUE	Yes	Set queue ID when outputting to a port
OFPAT_GROUP	Yes	Apply group
OFPAT_SET_NW_TTL	No	IP TTL
OFPAT_DEC_NW_TTL	Yes	Decrement IP TTL
OFPAT_SET_FIELD	Yes	Set a header field using IOXM TLV format
OFPAT_PUSH_PBB	No	Push a new PBB service tag (I-TAG)
OFPAT_POP_PBB	No	Pop the outer PP service tag (I-TAG)
OFPAT_EXPERIMENTER	No	Experimenter action

OpenFlow Supported Set Field Actions

TABLE 4 Set Field Actions

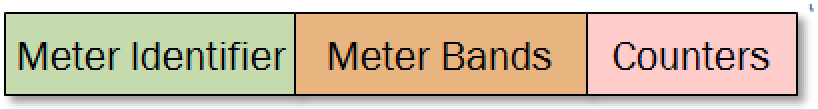
oxm_ofb_match_fields	ICX6610	Description
OXM_OF_ETH_DST	Yes	Modify Ethernet destination MAC address
OXM_OF_ETH_SRC	No	Modify Ethernet source MAC address
OXM_OF_ETH_TYPE	No	Modify Ethernet type of the OpenFlow packet payload, after VLAN tags
OXM_OF_VLAN_VID	Yes	Modify VLAN-ID
OXM_OF_VLAN_PCP	Yes	Modify VLAN-PCP
OXM_OF_IP_DSCP	Yes	Modify Diff Serv Code Point (DSCP). Part of the IPv4 ToS field
OXM_OF_IP_ECN	Yes	Modify ECN bits of the IP header

The definitions of the above mentioned match fields can be found in the OpenFlow 1.3 specification mentioned in the Reference Section later this document.

Meter Table

A meter measures the rate of packets assigned to it and enables controlling the rate of those packets either by lowering the priority of the packets or by dropping the packets that are received at a rate higher than the specified rate in the meter definition.

FIGURE 4 Structure of a Meter



A meter measures the rate of packets assigned to it and enables controlling the rate of those packets either by lowering the priority of the packets or by dropping the packets that are received at a rate higher than the specified rate in the meter definition.

Two types of bands namely DROP and DSCP are supported.

DROP band: Traffic exceeding the specified rate gets dropped.

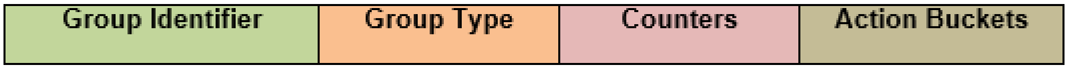
DSCP band: Traffic exceeding the specified rate is DSCP remarked to a higher drop precedence reducing its priority.

Only kbps is supported for packet rate specification. Packets-per-second (pps) is not supported

Group Table

A group table consists of group entries. The ability for a flow entry to point to a group enables OpenFlow to represent additional methods of forwarding.

FIGURE 5 OpenFlow 1.3 Group Entry



Group Identifier	Group Type	Counters	Action Buckets

Group Identifier: A 32 bit unsigned integer uniquely identifying the group.

Group Type: Determine the group semantics

Counters: Updated when packets are processed by a group

Action Buckets: Ordered list of action buckets, where each action bucket contains a set of

actions to execute and associated parameters

There are four group types that can be specified in the "Group Type" field: ALL, SELECT, INDIRECT,and Fast Failover

ALL - Executes all the buckets in the group.

SELECT - Select one bucket in the group using an algorithm and execute the selected bucket.

INDIRECT - Execute one defined bucket in the group.

Fast Failover - Execute the first live bucket.

Hybrid Switch Mode

By default, when the OpenFlow is enabled on the switch, the switch works in hybrid switch mode. In this mode, the ports enabled with OpenFlow mode only process the incoming traffic based on flows installed on the switch and the ports not enabled with OpenFlow mode continue to forward packets using the local forwarding engine. Either the port is an OpenFlow port or a non OpenFlow port. If no flow is installed on the OpenFlow port, by default the incoming packets will be dropped. User has the option to change this behavior by the means of a command.

Hybrid Port Mode

Hybrid port mode is introduced to enable the existing networks to slowly transition to OpenFlow without disruption to the existing configuration. Hybrid port mode works by letting the network administrators selectively choose the traffic that will be subject to OpenFlow flows while leaving the other incoming traffic untouched and routed through the traditional routing protocols. In the Brocade implementation of OpenFlow, this selection is made based on the VLANs configured on a port. These VLANs are protected and unprotected VLANs.

Protected VLAN : When configured, any incoming traffic with this VLAN tag will be subject to normal routing and is untouched by the OpenFlow flows.

Unprotected VLAN: When configured, the incoming traffic is matched against the existing flows. If a flow match is found, the associated action is taken. When no match is found, traffic is subject to normal routing.

OpenFlow Messages

OpenFlow messages are used for communication between the controller and the switch for establishing the connection, sending the flows, polling the flow/port statistics, and notifying the controller etc. Refer to the OpenFlow 1.3 Specification mentioned in the references section for a detailed explanation of different OpenFlow messages.

The table below shows the supported OpenFlow Messages on FastIron ICX 6610.

TABLE 5 Supported OpenFlow Messages

Message Type	Supported
OFPT_Hello	Yes
OFPT_ERROR	Yes
OFPT_ECHO_REQUEST	Yes
OFPT_REPLY	Yes
OFPT_EXPERIMENTER	No
OFPT_FEATURES_REQUEST	Yes
OFPT_FEATURES_REPLY	Yes
OFPT_GET_CONFIG_REQUEST	No
OFPT_CONFIG_REPLY	No
OFPT_SET_CONFIG	No
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

TABLE 5 Supported OpenFlow Messages (Continued)

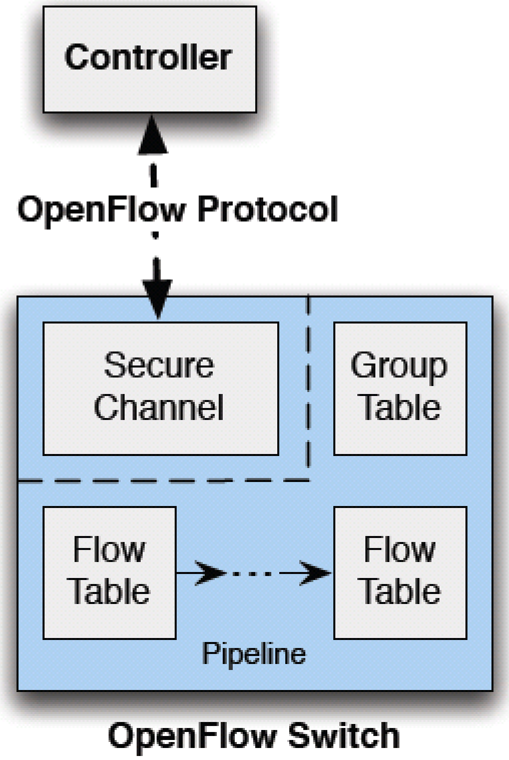
Message Type	Supported
OFPT_ROLE_REPLY	Yes
OFPT_GET_ASYNC_REQUEST	Yes
OFPT_GET_ASYNC_REPLY	Yes
OFPT_SET_ASYNC	Yes
OFPT_METER_MOD	Yes

OpenFlow Configuration

To verify the configuration, please see the section OpenFlow show Commands on page 16.

Adding the OpenFlow structure figure below for reference.

FIGURE 6 OpenFlow Structure



Enabling OpenFlow

To enable OpenFlow 1.3 on the switch:

```
BrocadeICX#Configure terminal
BrocadeICX(config)#Openflow enable ofv130
```

In the command shown above, use ofv100 instead of ofv130 for enabling OpenFlow 1.0.

Configuring the Default Behavior as Send-to-Controller

By default, when the incoming traffic does not match any of the installed flows, it gets dropped. User can choose to send this unknown traffic to the OpenFlow Controller in the form of Packet_IN messages.

```
BrocadeICX(config)#openflow default-behavior send-to-controller
BrocadeICX(config)#
```

Configuring a Connection to the Controller

The connection to the controller can be of two types:

- Active - The switch initiates the connection to the controller
- Passive - The switch passively listens on the specified port (6633 by default) for controller commands

It is up to the network administrator to choose between the Active and Passive modes. There is no added advantage to using one particular mode. The same applies for the connection type which can be TCP or SSL.

To configure passive connection with no ssl, enter the following command:

```
BrocadeICX(config)#Openflow controller passive no-ssl
```

To configure *Active connection* to the controller with IP address 10.20.226.189 , enter the following command:

```
BrocadeICX(config)#OpenFlow controller ip-address 10.20.226.189 no-ssl
```

Enabling OpenFlow on the Port

OpenFlow can be enabled on the port in three modesL:

- Layer2 - Match only layer2 fields in the data packet
- Layer3 - Match only layer3 and layer4 fields in the data packet
- Layer23 - Match layer2, layer3, and layer4 simultaneously

The following command shows how to enable OpenFlow on port 1/1/16 in layer2 mode. When OpenFlow is enabled on a port, the CDP/FDP protocols get disabled. A warning message is returned to inform the user about this change.

```
BrocadeICX(config)#interface ethe 1/1/16
BrocadeICX(config-if-e1000-1/1/16)#OpenFlow enable layer2
Warning: Enabling OpenFlow on port e1/1/16 disables CDP/FDP and spanning-tree on it
BrocadeICX(config-if-e1000-1/1/16)#
```

In the command shown above, use 'layer3' or 'layer23' instead of 'layer2' to enable OpenFlow on the interface in layer3 or layer23 mode respectively. In Layer2 mode and Layer3 mode, each flow occupies one TCAM rule whereas Layer23 mode occupies two TCAM rules.

Configuring Protected/Unprotected VLANs for Hybrid Mode (optional)

When OpenFlow is enabled in Hybrid-Port mode on a port, the user has the option to configure protected or unprotected VLANs. By default, any VLAN that this port is tagged in is treated as an unprotected VLAN. No additional configuration is necessary. Protected VLANS however, need to be explicitly mentioned using the command shown below so that they are not treated as unprotected VLANs.

To configure protected VLANs 100 and 200 on port 1/1/17, use the following procedure:

1. Enable OpenFlow on port 1/1/17 in layer2 hybrid port mode.
   ```
   BrocadeICX(config-if-e1000-1/1/17)# openflow enable layer2 hybrid-mode
   Warning: Enabling OpenFlow on port e1/1/17 disables CDP/FDP, MAC-learning and
   spanning-tree on it
   ```
2. Configure protected VLANs 100 and 200.
   ```
   BrocadeICX(config-if-e1000-1/1/17)#openflow protected-vlans 100 200
   BrocadeICX(config-if-e1000-1/1/17)#
   ```
3. Tag the port 1/1/17 to VLANS 100, 200, and 300. VLAN 300 will be an unprotected VLAN.

NOTE
It is mandatory to tag the port to the VLANs configured as protected VLANs.

```
BrocadeICX(config-vlan-10)#vlan 100
BrocadeICX(config-vlan-100)#tagged ethe 1/1/17
Added tagged port(s) ethe 1/1/17 to port-vlan 100.
BrocadeICX(config-vlan-100)#vlan 200
BrocadeICX(config-vlan-200)#tagged ethe 1/1/17
Added tagged port(s) ethe 1/1/17 to port-vlan 200.
BrocadeICX(config-vlan-200)#vlan 300
BrocadeICX(config-vlan-300)#tagged ethe 1/1/17
Added tagged port(s) ethe 1/1/17 to port-vlan 300.
BrocadeICX(config-vlan-300)#
```

Configuring System-max Settings for OpenFlow

The following system max settings are required when enabling OpenFlow in scaled scenarios:

- OpenFlow Flow Entries (needs a reboot):

  The following command configures the system-max setting for OpenFlow entries to 3000.
  ```
  BrocadeICX(config)#system-max openflow-flow-entries ?
    DECIMAL   Valid range 0 to 12000 (default: 1024)
  BrocadeICX(config)#system-max openflow-flow-entries 3000
  Reload required.  Please write memory and then reload or power cycle.
  BrocadeICX(config)#
  ```

NOTE
While the available range for this command is 0 to 12000, it should be kept in mind that only 3000 flows can be configured per device(or Packet Processor). The different number of maximum supported flows is mentioned in the table below.

TABLE 6 Maximum Number of Supported Flows

ICX6610 model	No. of Devices	No. of Flows Supported in Layer23 Mode per Device	No. of Flows Supported in Layer2 or Layer3 Mode per Device	No. of Flows supported in Layer3 Mode with IPv6 Address Matching per Device
ICX6610-24/24P	1	1500	3000	1500
ICX6610-48/48P	2 (Ports 1 to 24 are on device 1; ports 25-48 are on device 2)	1500	3000	1500

Based on the number of ICX6610 units that are configured in a stack, a user can chose to set this value to a higher number. For example in a two unit ICX6610-24 stack, a user can set this value to

6000 (3000x2). In a two unit ICX6610-48 stack, a user can set this value to 12000 because it has four available devices.

- OpenFlow Group Entries for Group Type SELECT (needs a reboot):

  The following command configures the system-max setting maximum number of group entries defined in SELECT mode. This in turn reduces the number of Link Aggregation Groups (LAGs) available in the system that are used for normal forwarding. The number of LAGs will be 127 minus the number of group SELECT entries configured. This setting is only applicable for OpenFlow groups defined in SELECT mode. The other modes ALL, INDIRECT and FF (Fast Failover) are not impacted by this setting.

```
BrocadeICX(config)#system-max openflow-group-select-entries  ?
  DECIMAL   Valid range 0 to 120 (default: 0)
BrocadeICX(config)#system-max openflow-group-select-entries 100
Reload required.  Please write memory and then reload or power cycle.
BrocadeICX(config)#
```

- OpenFlow Next Hop Entries for the TTL Decrement Action (needs a reboot):

  The following command shows the system-max value for OpenFlow next-hop entries set to 500. If the number of flows that have the action to decrement the network TTL go beyond the number specified in this command, they will be rejected.

```
BrocadeICX(config)#system-max openflow-nexthop-entries ?
  DECIMAL   Valid range 0 to 1024 (default: 0)
BrocadeICX(config)#system-max openflow-nexthop-entries 500

Total max configured ipv4 routes are 12000
  - Max ipv4 routes configured for default VRF are 7000
  - Max openflow nexthop entries configured are 500(route used 2000)
  - Max ipv4 routes available for all non-default VRFs are 3000
Warning: Please revalidate these values to be valid for your configuration.
Reload required.  Please write memory and then reload or power cycle.
BrocadeICX(config)#
```

- OpenFlow Protected/Unprotected VLAN Entries (needs a reboot):

  The following command shows the system max values for protected VLANs and unprotected VLANS for the system set to 200 each.

```
BrocadeICX(config)#system-max openflow-pvlan-entries
  DECIMAL   Valid range 0 to 256 (default: 40)
BrocadeICX(config)#system-max openflow-pvlan-entries 200
Reload required.  Please write memory and then reload or power cycle.
BrocadeICX(config)#system-max openflow-un
  openflow-unprotectedvlan-entries
BrocadeICX(config)#system-max openflow-unprotectedvlan-entries
  DECIMAL   Valid range 0 to 256 (default: 40)
BrocadeICX(config)#system-max openflow-unprotectedvlan-entries 200
Reload required.  Please write memory and then reload or power cycle.
BrocadeICX(config)#
```

- **NOTE**

  These protected/unprotected VLAN settings are for the entire switch. On any given port, a user can only configure a maximum of 40 protected/unprotected VLANs.

OpenFlow show Commands

Display OpenFlow Information

The following output displays the OpenFlow information including the version, configured controllers, match capability, default behavior, and OpenFlow interfaces.

```
BrocadeICX#show openflow
Administrative Status:       Enabled
SSL Status:                  Enabled
Controller Type:             ofv130 <- Indicates the OpenFlow version
Number of Controllers:       2<- Shows the total number of Controller connections
Controller 1:
Connection Mode:       passive, TCP <-Shows the connection mode for Controller 1
Listening Address:     0.0.0.0
Connection Port:       6633 <- Shows the port on which the switch is listening for
```

```
controller
Connection Status:       TCP_LISTENING
Role:                    EQUAL <- Role of the controller.
Controller 2:
Connection Mode:        active, TCP <-Shows the connection mode for Controller 2
Controller Address:      10.20.226.189 <-Shows the IP address of the controller
Connection Port:        6633
Connection Status:       Established <-Shows the connection status
Role:                    EQUAL
Packet-in config:        no-match, action
Port-status config:      add, delete, modify
Flow-removed config:     delete, group-delete

Match Capability: <- Shows the Matching capabilities of the switch
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:      e1/1/1 e1/1/9 e1/1/11 e1/1/37 e1/1/38 e1/1/47
                                    e1/3/3 e1/3/4 e1/3/7 e1/3/8 <-Displays OpenFlow
interfaces
Openflow Hybrid Interfaces: <- Displays interfaces in hybrid port mode
e1/1/17
Protected VLANs   :   100,  200, <- Protected VLANs
Unprotected VLANs :    300, <-shows the unprotected VLANs
Default action: send-to-controller <- Default Action configured
Maximum number of flows allowed: 12000
Active flow: 1 – Total number of flows installed.
Maximum number of Protected Vlans allowed: 256
Maximum number of Unprotected Vlans allowed: 256
Total number of Protected Vlans: 2
Total number of Unprotected Vlans: 1
```

Display OpenFlow Interfaces

The following command displays the list of OpenFlow interfaces and the mode in which OpenFlow is configured on them.

OF-portid: The output below indicates the OpenFlow port id which is in decimal format.

```
BrocadeICX#show openflow interface

Total number of Openflow interfaces: 11

Port   Link   Speed Tag MAC            OF-portid   Name           Mode
1/1/1 Disable None  No  748e.f894.07c6 1                          Layer3
1/1/9  Down   None  No  748e.f894.07ce 9                          Layer2
1/1/11 DisableNone  No  748e.f894.07d0 11                         Layer2
1/1/16 Down   None Yes 748e.f894.07d5 16                         Hybrid-Layer2
1/1/17 Up     1G    No  748e.f894.07ea 17                         Layer2
1/1/38 DisableNone  No  748e.f894.07eb 38                         Layer2
1/1/47 Up     1G    No  748e.f894.07f4 47                         Layer23
1/3/3  Up     None  No  748e.f894.0803 131                        Layer23
1/3/4  DisableNone  No  748e.f894.0804 132                        Layer23
1/3/7  Up     10G   No  748e.f894.0807 135                        Layer2
1/3/8  DisableNone  No  748e.f894.0808 136                        Layer2
```

Display OpenFlow Controller Connection Status

The following command displays the list of connections configured to the controllers.

```
BrocadeICX# show openflow controller
Openflow controller information
--------------------------------------------------------------------------------
  Controller   Mode      TCP/SSL   IP-address        Port   Status
--------------------------------------------------------------------------------
  1  (Equal)   passive   TCP       0.0.0.0           6633   TCP_LISTENING
  2  (Equal)   active    TCP       10.20.226.189     6633   OPENFLOW_ESTABLISHED
BrocadeICX#
```

The "status" column shows the connection status between switch and the controller.

For the connection that is in passive mode, the status will be shown as "TCP_LISTENING" for most of the time. When a controller establishes connection and sends the flows, the status will be updated to "OPENFLOW_ESTABLISHED" for the duration of the connection.

For the connection in Active mode, the status can be "TCP_CONNECTING" when the switch is attempting to initiate a session with the controller, "OPENFLOW_ESTABLISHED" when the connection is successfully established, and "CLOSED" when the connection is closed.

Display OpenFlow Flows

The following command displays the list of OpenFlow flows installed on the switch.

```
BrocadeICX#show openflow flows
Total Number of data packets sent to controller:                    8
Total Number of data bytes sent to controller  :                 4752
Total Number of Flows: 2
        Total Number of Port based Flows: 1
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 1
        Total Number of L23 Generic Flows: 0
Total Number of Hardware entries for flows: 11
        Total Number of Hardware entries for Port flow: 1
        Total Number of Hardware entries for Generic flow: 10
Total Number of Openflow interfaces: 11
        Total Number of L2  interfaces: 7
        Total Number of L3  interfaces: 1
        Total Number of L23 interfaces: 3
Flow ID: 1 Priority: 0 Status: Active
        Rule:
           In Port:      generic <- shows the input port.
        Instructions: Apply-Actions
                Action: FORWARD <- shows the actions associated with the
flow
                        Out Port: send to controller
        Statistics:  <-Shows the total number of packets that match this flow.
           Total Pkts: 0
           Total Bytes: 0
Flow ID: 2 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/3/3
           Ether type:   0x800
           IP Protocol:          17
           IP Protocol Source Port:      100
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/3/4
   Output port:  1/3/4
        Statistics:
           Total Pkts: 101
           Total Bytes: 6466
BrocadeICX#
```

Display OpenFlow Groups

The following command shows the Groups defined on the switch.

```
BrocadeICX#show openflow group

Max number of groups            : 512
Max number of buckets per group  :  64
Max number of actions per bucket :   1

Max number of SELECT groups          :  100 <- This value is based on the system-max
setting.
Max number of buckets in SELECT group:   8
Starting Trunk ID for SELECT groups  :  71

Group id 1  shows the group ID (set by the controller)

  Transaction id        4043243760
  Type                  SELECT <- shows the group type (Can be ALL, INDIRECT,FF or
SELECT)
  Packet Count          0
  Byte   Count          0
  Flow   Count          1
  Number of buckets     3 <- Specifies the number of Action buckets within the group

  bucket #1
    Weight             50 <- Weights are not supported. 50 is picked by default and
has



                                                  no meaning
    Number of actions   1 <- This can only be 1 .If more than one actions per action
bucket

                              are sent from controller, the group will be rejected by
the switch
      action 1: out port: 1/1/37 <- Shows the output port.

  bucket #2
    Weight             50
    Number of actions   1
      action 1: out port: 1/1/38

  bucket #3
    Weight             50
    Number of actions   1
      action 1: out port: 1/3/7
   Forwarding information:
  LTM Index: 0   Trunk-Id: 71

----
```

Display OpenFlow Meters

The following command shows the meters configured on the switch.

```
BrocadeICX#show openflow meters

TOTAL Meters in the system: 1

Meter id: 1 <-

  Transaction id:        4043243760
  Meter Flags:            KBPS BURST <- Shows the packet rate measurement criterion.
Only



                          KBPS is supported.
  Flow Count:            1<- shows the number of flows this meter is attached to.
  Number of  bands:      1<- shows the number of bands defined in the meter
  In packet count:       -NA-
  In byte count:         0

  Band Type:    DROP <- Shows the type of the band (can be DROP or DSCP)

    Rate:                        170000 <-shows the rate (in KBPS)
    Burst size:                  1250          kb<- shows the burst size.
  In packet band count:         -NA- <- Shows the number of packets measured in this
band.
    In byte band count:          0

Total no. of entries printed: 1

BrocadeICX#
```

OpenFlow Controllers

The following controllers were used for testing purposes:

- Brocade Vyatta Controller
- Open Daylight - version 0.0.4
- OVS-OFCTL - version 2.0.0

- DPCTL - version 1.3.0
- MUL - version 3.2.7
- SPIRENT/IXIA

Use Cases

The following equipment is used:

- ICX6610 running software version 08.0.20
- MLXe-4 running version 05.7.00
- Ubuntu Linux PC (version 12.0.4) for running the OVS controller
- Spirent Test Center traffic generator

For both of these use cases, the following needs to be enabled before proceeding with the test.

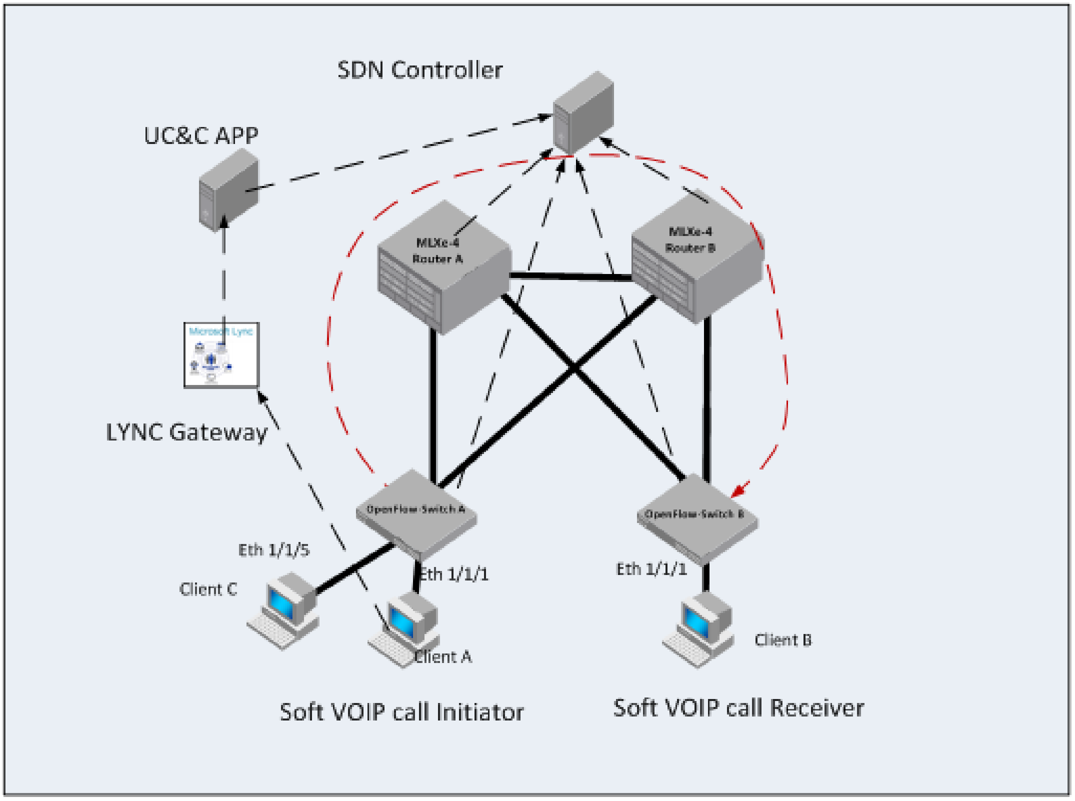
1. Enable OpenFlow 1.3
```
BrocadeICX#Configure terminal
BrocadeICX(config)#Openflow enable ofv130
```
2. Configure a passive controller connection
```
BrocadeICX#Configure terminal
BrocadeICX(config)#Openflow controller passive no-ssl
```

For verification details, see the section OpenFlow show Commands on page 16.

Application-based Resource Allocation

Consider the topology below where the work stations Client A and Client B are connected via the access switches switch A and switch B. The network comprises of MLX3 routers at the aggregation/ core. OpenFlow is enabled on all the switches.

FIGURE 7 Application-based Resource Allocation



Consider the case where the Lync application on client A initiates a call to the Lync application on Client B. Client A contacts the LYNC call gateway and gets a response in the form of Client B's IP address following which the call is established. In the present day deployments, prioritizing the VOIP traffic generated by this VOIP call gets tricky. This is because the prioritization rules like access-lists for example need to be deployed across all the switches/routers in the path. Given the dynamic nature of these IP addresses of clients (especially wireless clients), updating the rules across all the network nodes is a cumbersome task.

In the case of SDN, since all the nodes within the enterprise are OpenFlow-enabled, a single point of automated control makes it easier to prioritize traffic with flows and manage the network efficiently. For example, consider all the nodes shown in the topology above are OpenFlow-enabled. The VOIP call traffic can be prioritized in the following way.

1. Client A initiates the VOIP call by contacting the LYNC call controller. By this time, Client C is already communicating with Client B for accessing the FTP service.
2. Consider the normal data traffic between Client A and Client B follows the path1 (shown in RED dotted lines).
3. The LYNC call controller responds back to Client A with Client B's IP address.
4. In addition, The LYNC controller also informs the UC-Collaboration (UC&C) app about the VOIP call by providing the IP addresses of both Client A and Client B.
5. The UC&C app then instructs the controller to push flows to prioritize the traffic between the IP pair of Client A and Client B that are destined to a certain TCP/UDP port (TCP port 5063 for example).
6. The SDN controller immediately pushes the flows to the switches A and B to enqueue the matching traffic in a higher queue on the output port so that it get priority over other traffic.

   With the push of a few flows, the network administrator is able to achieve low latency and prioritization for VOIP calls .

Note: Since the app mentioned is not available, flows are manually sent from the controller to mimic the events mentioned above

Topology

Client A is connected to port 1/1/1 on SwitchA . Client C is connected to port 1/1/5 on Switch A. Switch A has two outgoing ports 1/1/2 and 1/1/3 towards MLXe-A and MLXe-B respectively.

Client B is connected to port 1/1/ on Switch B. Switch B has two outgoing ports 1/1/2 and 1/1/3 connected to MLXe-B and MLXe-A respectively.

Configuration

1. Enable OpenFlow layer23 mode on all the ports on switch A.

```
switchA(config)#int e 1/1/1 to 1/1/5
switchA(config-mif-1/1/1-1/1/5)#openflow enable layer23
Warning: Enabling OpenFlow on port e1/1/1 disables CDP/FDP and spanning-tree on it
Warning: Enabling OpenFlow on port e1/1/2 disables CDP/FDP and spanning-tree on it
Warning: Enabling OpenFlow on port e1/1/3 disables CDP/FDP and spanning-tree on it
switchA(config-mif-1/1/1-1/1/5)#
switchA(config-mif-1/1/1-1/1/5)#
switchA(config-mif-1/1/1-1/1/5)#exit
switchA(config)#
```

2. Enable OpenFlow layer23 mode on all the ports on switch B.

```
switchB(config-mif-1/1/1-1/1/3)#openflow enable layer23
Warning: Enabling OpenFlow on port e1/1/1 disables CDP/FDP and spanning-tree on it
Warning: Enabling OpenFlow on port e1/1/2 disables CDP/FDP and spanning-tree on it
Warning: Enabling OpenFlow on port e1/1/3 disables CDP/FDP and spanning-tree on it
switchB(config-mif-1/1/1-1/1/3)#
switchB(config-mif-1/1/1-1/1/3)#
switchB(config-mif-1/1/1-1/1/3)#exit
switchB(config)#
```

   Similarly, configure OpenFlow on ports of MLXe-A and MLXe-B.

3. Send flows from the controller to Switch A instructing to forward any traffic received on port 1/1/1 and port 1/1/5 toward MLXe-A (port 1/1/2).

```
OVS Controller:
The following command sends the flow for traffic coming in on port 1/1/1 to be
sent out on
port 1/1/2. Ethernet type 800 is mentioned to specify the traffic is IPv4.
brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.102  -O
OpenFlow13 --flow-format OXM  "in_port=1 dl_type=0x800 action=output:2"

The following command sends the flow for traffic coming in on port 1/1/5 to be
sent out on
port 1/1/2. Ethernet type 800 is mentioned to specify the traffic is IPv4.
brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.102  -O
OpenFlow13 --flow-format OXM  "in_port=5 dl_type=0x800 action=output:2"



Verify flows on the switch:

switchB#show openflow flows
Total Number of data packets sent to controller:                1184
Total Number of data bytes sent to controller  :              120573

Total Number of Flows: 2 <- Number of flows is shown as 2
        Total Number of Port based Flows: 2
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 0
        Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 12
        Total Number of Hardware entries for Port flow: 2
        Total Number of Hardware entries for Generic flow: 10

Total Number of Openflow interfaces: 12
        Total Number of L2  interfaces: 8
        Total Number of L3  interfaces: 0
        Total Number of L23 interfaces: 4


Flow ID: 1 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/5 <- Input port 1/1/5
           Ether type:   0x800
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2 <- Output port 1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 2017
           Total Bytes: 120586
Flow ID: 1 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/5
           Ether type:   0x800
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

4. Send flow from the controller to Switch B instructing the controller to forward any traffic received on port 1/1/1 towards MLXe-B (port 1/1/2).

```
OVS Controller:
The following command sends the flow for traffic coming in on port 1/1/1 to be
sent out on
port 1/1/5. Ethernet type 800 is mentioned to specify the traffic is IPv4.
brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.142  -O
OpenFlow13 --flow-format OXM  "in_port=1 dl_type=0x800 action=output:2"

switchB#show openflow flows
Total Number of data packets sent to controller:                 1256
Total Number of data bytes sent to controller  :               145528

Total Number of Flows: 1
        Total Number of Port based Flows: 1
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 0
        Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 6
        Total Number of Hardware entries for Port flow: 1
        Total Number of Hardware entries for Generic flow: 5

Total Number of Openflow interfaces: 5
        Total Number of L2  interfaces: 1
        Total Number of L3  interfaces: 0
        Total Number of L23 interfaces: 4


Flow ID: 1 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/1 <- Input port is 1/1/1
           Ether type:   0x800
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2 <- Output port is 1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 5158
           Total Bytes: 304586

switchB#
```

5. Using simulated Client C (traffic generator), initiate traffic to simulated Client B (traffic generator).

   Use 950 Mbps of traffic rate (~95% of 1G port)

   Check the statistics on switch A and switch B to verify the traffic is received on the correct ports.

```
Switch A:
switchA# show statistics brief ethe 1/1/2 ethe 1/1/5

Port             In Packets       Out Packets       In Errors      Out Errors
```

```
1/1/2                     0                 5556               0               0
1/1/5                     5557                0              0               0

TOTAL                     5557                5556               0               0

Check if the number ‘In packets’ on port 1/1/5 match ‘Out packets’ on port
1/1/2


Switch B:
switchB#show statistics brief eth 1/1/1 e 1/1/2

Port             In Packets       Out Packets       In Errors      Out Errors
1/1/1                     6212                 0              0               0
1/1/2                     0                 6212              0               0

TOTAL                     6212                6212        0               0
Check if the number ‘In packets’ on port 1/1/1 match ‘Out packets’ on port
1/1/2
```

6. Using simulated Client A (traffic generator), initiate traffic to simulated Client B (traffic generator) with source IP address of 1.1.1.1 and destination IP address of 2.1.1.1 and TCP port 5063.

   Use 85Mbps (~8.5% of 1G) oversubscribing the output port 1/1/2 to 103.5%.

   Check the interface statistics on port 1/1/2 of switch A.

```
->Check the interface to see which egress queue the traffic is sent out on port
1/1/2

switchA#show interface ethe 1/1/2
GigabitEthernet1/1/2 is up, line protocol is up
  Port up for 1 hours 12 minutes 30 seconds
  Hardware is GigabitEthernet, address is 748e.f894.07ea (bia 748e.f894.07ea)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled, Designated protect is Disabled
  Link Error Dampening is Disabled
  STP configured to OFF, priority is level0, mac-learning is enabled
  Openflow is Enabled, Openflow Hybrid mode is Disabled, Flow Type is Layer23
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor enabled (output only)
  Mac-notification is disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  MTU 1500 bytes, encapsulation ethernet
  300 second input rate: 200 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 104757598 bits/sec, 212262 packets/sec, 100.00%
utilization
                                    <-Shows the 100% output utilization on the
port.
  1013 packets input, 113026 bytes, 0 no buffer
  Received 3 broadcasts, 1010 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  212262 packets output, 1698098 bytes, 0 underruns
  Transmitted 7 broadcasts, 1015 multicasts, 0 unicasts
  0 output errors, 0 collisions
  Relay Agent Information option: Disabled

Egress queues: <- shows the egress queue statistics
Queue counters    Queued packets    Dropped Packets
    0                   35563               25569 <-Drop
    1                   0                   0
    2                   0                   0
    3                   0                   0
    4                   0                   0
    5                   5                   0
    6                   0                   0
    7                   1                   0
Verify if the traffic is egressing on queue 0 with excessive traffic being
dropped.
```

7. Now send a flow from SDN controller to switch A with the match for traffic with source IP 1.1.1.1 and destination IP 2.1.1.1 plus the TCP destination port 5063. The action must be to output the traffic on port 1/1/2 in the queue.

```
OVS controller:
The following command sends the flow for traffic coming in on port 1/1/1 with
source IP
address 1.1.1.1 ,destination IP address 2.1.1.1 and TCP destination port 5063 to
be sent
out on port 1/1/2 in queue 5. Ethernet type 800 is mentioned to specify the
traffic is IPv4.
brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.102  -O
OpenFlow13 --flow-format OXM  "in_port=1 dl_type=0x800 nw_src=1.1.1.1
nw_dst=2.1.1.1
nw_proto=17 tp_src=100 action=enqueue:2:5 output:2"

On the switch:
switchA#show openflow flow
Total Number of data packets sent to controller:                 1850
Total Number of data bytes sent to controller  :               188207

Total Number of Flows: 2
        Total Number of Port based Flows: 2
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 1
        Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 12
        Total Number of Hardware entries for Port flow: 2
        Total Number of Hardware entries for Generic flow: 10

Total Number of Openflow interfaces: 12
        Total Number of L2  interfaces: 8
        Total Number of L3  interfaces: 0
        Total Number of L23 interfaces: 4
Flow ID: 2 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/5
           Ether type:   0x800
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 500524
           Total Bytes: 3642588

Flow ID: 3 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/1
           Ether type:   0x800
           Source IP:            1.1.1.1        Subnet IP:       255.255.255.255
           Destination IP:       2.1.1.1        Subnet IP:       255.255.255.255
           IP Protocol:          17
           IP Protocol Destination Port:      5063
        Instructions:Apply-Actions
                Action: FORWARD
            Out Port:e 1/1/2, Queue: 5
        Statistics:
           Total Pkts: 5568
           Total Bytes: 32478
```

8. Check the port statistics of port 1/1/2 on switch A.

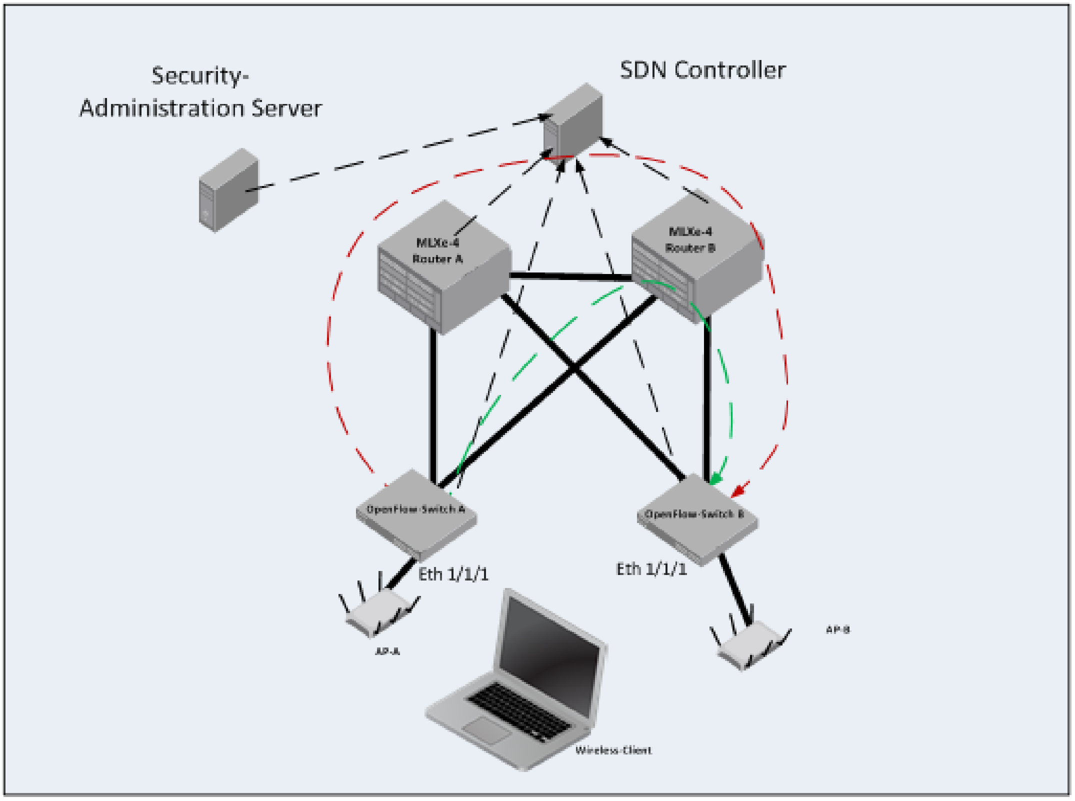
```
->Check the interface to see which egress queue the traffic is sent out on port
1/1/2

switchA#show interface ethe 1/1/2
GigabitEthernet1/1/2 is up, line protocol is up
  Port up for 1 hours 12 minutes 30 seconds
  Hardware is GigabitEthernet, address is 748e.f894.07ea (bia 748e.f894.07ea)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled, Designated protect is Disabled
  Link Error Dampening is Disabled
  STP configured to OFF, priority is level0, mac-learning is enabled
  Openflow is Enabled, Openflow Hybrid mode is Disabled, Flow Type is Layer23
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor enabled (output only)
  Mac-notification is disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  MTU 1500 bytes, encapsulation ethernet
  300 second input rate: 200 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 104757598 bits/sec, 212266 packets/sec, 100.00%
utilization
  1013 packets input, 113026 bytes, 0 no buffer
  Received 3 broadcasts, 1010 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  2122622 packets output, 12898098 bytes, 0 underruns
  Transmitted 7 broadcasts, 1015 multicasts, 0 unicasts
  0 output errors, 0 collisions
  Relay Agent Information option: Disabled

Egress queues:
Queue counters    Queued packets    Dropped Packets
    0                   32158              24588 <- Excessive non-lync traffic
dropped.
    1                   0                   0
    2                   0                   0
    3                   0                   0
    4                   0                   0
   5                   3501                0 <- LYNC traffic is prioritized in
queue5 with no drops
    6                   0                   0
    7                   1                   0
```

It can be seen that the VOIP traffic is now sent out on Queue 5 avoiding the DROP.

Role-based Access Control

Consider the case where the MAC address of a client is pre-provisioned (or dynamically in case of RADIUS authentication) by a Security Administration server.

FIGURE 8 Role-based Access Control



Consider the case where a client (MAC address) has access to resources in one part of the network, but should be restricted in other parts of network where the client is not authorized.

An Example Scenario

A student in a campus is network trying to access contents of the network while at library (switch A). The moment the student's laptop enters another building (networking lab), the student's access should be restricted to only the internet, but not the building's network.

The following steps would explain this scenario:

1. Install a generic flow on switch A and switch B to send any unknown packets to the controller.
2. Run a DHCP service on the SDN controller server.
3. When the student connects to library's wireless network via access point A, the controller upon verifying the Mac address with the security server responds to the wireless client's DHCP request with an IP address. Since the student is allowed to access the library's local network as well as the internet, a simple flow is installed on switch A, allowing the student access.
4. Now, when the student moves to another building where the networking lab is, the student's access should only be granted to use the local network but not the internet. The SDN controller now pushes a flow to switch B instructing to add flows that drop the wireless-client (student)'s traffic to restricted network addresses while allowing traffic to local networks.

NOTE
Static IP address is used in this case, since a DHCP server that can be run with an SDN controller is not available. In addition, this scenario assumes that the MAC address of the client is already assumed to be configured for permission to access/restrict certain networks (library and network lab in this case). Wired connection is used instead of the wireless.

Configuration

1. Enable OpenFlow using the steps mentioned in scenario 1.
2. Configure default behavior on switches A and B to send unknown packets to controller.

   ```
   switchA(config)#openflow default-behavior send-to-controller
   switchA(config)#
   switchB(config)#openflow default-behavior send-to-controller
   switchB(config)#
   ```

   Verification of this step can only be done by sniffing the packets coming to the controller since it is a simulated controller. Look for "PACKET_IN type message".
3. Assume the wireless client connects to wireless AP-A. The SDN controller should push a flow to forward traffic from the MAC address of the client to all networks.

   ```
   OVS controller:
   The following command send a flow for traffic coming in on port 1/1/1  with MAC
   address of the
   wireless client (PC) to be forwarded out on port 1/1/2.
   brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.102  -O
   OpenFlow13 --
   flow-format OXM  "in_port=1 dl_src=01:01:01:55:63:12 output:2"

   On Switch A:
   switchA#show openflow flow
   Total Number of data packets sent to controller:                2903
   Total Number of data bytes sent to controller  :              295140

   Total Number of Flows: 1
           Total Number of Port based Flows: 1
           Total Number of L2 Generic Flows: 0
           Total Number of L3 Generic Flows: 0
           Total Number of L2+L3 Generic Flows: 0
           Total Number of L23 Generic Flows: 0

   Total Number of Hardware entries for flows: 7
           Total Number of Hardware entries for Port flow: 1
           Total Number of Hardware entries for Generic flow: 6

   Total Number of Openflow interfaces: 12
           Total Number of L2  interfaces: 11
           Total Number of L3  interfaces: 0
           Total Number of L23 interfaces: 1

   Flow ID: 4 Priority: 32768 Status: Active
           Rule:
              In Port:      e1/1/1
              Source Mac:   748e.0155.6312
              Source Mac Mask:      ffff.ffff.ffff
           Instructions: Apply-Actions
                   Action: FORWARD
                           Out Port:  e1/1/2
      Output port:  1/1/2
           Statistics:
              Total Pkts: 556
              Total Bytes: 4568 <-This count indicates that the traffic is getting
   forwarded
   ```
4. Assume the wireless client moves to AP-A. The SDN controller should push a flow to restrict traffic to the internet. The local network is configured in the range 10.0.0.0/8. From the traffic generator, send two traffic streams destined to 1.1.1.1 and 10.0.0.1 each at 1000 packets/sec.

   ```
   OVS controller:
   The following command send a flow to switch B for traffic coming in on port
   1/1/1  with MAC
   ```

address of the wireless client (PC) and destination IP of 10.0.0.0/8 to be forwarded out on
port 1/1/2. In addition, disable the default behavior configured earlier.

```
brocade@brocade-ecmp:/mnt/windows/keys$ ovs-ofctl add-flow tcp:10.20.226.142  -O
OpenFlow13
--flow-format OXM  "in_port=1 dl_src=74:8e:01:55:63:12 dl_type=0x800
nw_dst=10.0.0.0[255.0.0.0]
action=output:2"
```

Switch B:
Remove the default behavior configured earlier.

```
SwitchB (config)#no openflow default-behavior send-to-controller
SwitchB(config)#
```

Check the OpenFlow flows

```
switchB#show openflow flows
Total Number of data packets sent to controller:                 3098
Total Number of data bytes sent to controller  :               314955

Total Number of Flows: 2
        Total Number of Port based Flows: 2
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 0
        Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 8
        Total Number of Hardware entries for Port flow: 2
        Total Number of Hardware entries for Generic flow: 6

Total Number of Openflow interfaces: 12
        Total Number of L2  interfaces: 8
        Total Number of L3  interfaces: 0
        Total Number of L23 interfaces: 4

Flow ID: 5 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/1
           Source Mac:   748e.0155.6312
           Source Mac Mask:      ffff.ffff.ffff
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 0
           Total Bytes: 0

Flow ID: 6 Priority: 32768 Status: Active
        Rule:
           In Port:      e1/1/1
           Source Mac:   748e.0155.6312
           Source Mac Mask:      ffff.ffff.ffff
           Ether type:   0x800
           Destination IP:       10.0.0.0       Subnet IP:       255.255.255.255
                     It can be seen that the flow allowing only 10.0.0.0/8 subnet
got installed.
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1/2
   Output port:  1/1/2
        Statistics:
           Total Pkts: 554
           Total Bytes: 556
Check the Statistics on switch B Switch B:
switchB#show statistics brief eth 1/1/1 e 1/1/2

Port             In Packets       Out Packets       In Errors      Out Errors
1/1/1                    6102                0              0              0
1/1/2                    0               3048               0              0

TOTAL                    6212              3048        0              0
```

The traffic coming out of port 1/1/2 is approximately half of the traffic coming in on port 1/1/1. A centralized controller makes the access control user-friendly for administrators and avoids human errors while configuring numerous devices in the network.

References

1. FI SDN configuration guide for 08.0.20 release
2. Software Defined Networking in the Campus Network:

   http://www.brocade.com/forms/getFile?p=documents/white_papers/brocade-sdn-in-campus-networks-wp.pdf
3. OpenFlow 1.3 Specification:

   https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf
4. OVS command reference

   http://www.pica8.com/document/v2.3/html/ovs-commands-reference/