Black Book

ixia

Edition 10

SDN/OpenFlow

http://www.ixiacom.com/blackbook

June 2014

Your feedback is welcome

Our goal in the preparation of this Black Book was to create high-value, high-quality content. Your feedback is an important ingredient that will help guide our future books.

If you have any comments regarding how we can improve the quality of this book, or suggestions for topics to be included in future Black Books, please contact us at ProductMgmtBooklets@ixiacom.com.

Your feedback is greatly appreciated!

Contents

How to Read this Book vii
Dear Reader viii
Introduction – Software Defined Networking 1
OpenFlow 2
OpenFlow Basic 3
Test Case: OpenFlow Switch Setup and Functional Test 7
Test Case: OpenFlow Switch Forwarding Test 21
Test Case: Switch Flow Failover Performance Test 43
Test case: OpenFlow Controller Scalability Test 63
Test case: Packet_out Rate Calculation 75
Test Case: Bandwidth Rate Limiting and QoS validation 85
Contact Ixia 103

How to Read this Book

The book is structured as several standalone sections that discuss test methodologies by type. Every section starts by introducing the reader to relevant information from a technology and testing perspective.

Each test case has the following organization structure:

Overview	Provides background information specific to the test case.
Objective	Describes the goal of the test.
Setup	An illustration of the test configuration highlighting the test ports, simulated elements and other details.
Step-by-Step Instructions	Detailed configuration procedures using Ixia test equipment and applications.
Test Variables	A summary of the key test parameters that affect the test's performance and scale. These can be modified to construct other tests.
Results Analysis	Provides the background useful for test result analysis, explaining the metrics and providing examples of expected results.
Troubleshooting and Diagnostics	Provides guidance on how to troubleshoot common issues.
Conclusions	Summarizes the result of the test.

Typographic Conventions

In this document, the following conventions are used to indicate items that are selected or typed by you:

- **Bold** items are those that you select or click on. It is also used to indicate text found on the current GUI screen.
- *Italicized* items are those that you type.

Dear Reader

Ixia's Black Books include a number of IP and wireless test methodologies that will help you become familiar with new technologies and the key testing issues associated with them.

The Black Books can be considered primers on technology and testing. They include test methodologies that can be used to verify device and system functionality and performance. The methodologies are universally applicable to any test equipment. Step-by-step instructions using Ixia's test platform and applications are used to demonstrate the test methodology.

This tenth edition of the black books includes twenty two volumes covering key technologies and test methodologies:

Volume 1 – Higher Speed Ethernet

Volume 2 – QoS Validation

Volume 3 – Advanced MPLS

Volume 4 – LTE Evolved Packet Core

Volume 5 – Application Delivery

Volume 6 – Voice over IP

Volume 7 – Converged Data Center

Volume 8 – Test Automation

Volume 9 – Converged Network Adapters

Volume 10 – Carrier Ethernet

Volume 11 – Ethernet Synchronization

Volume 12 – IPv6 Transition Technologies

Volume 13 – Video over IP

Volume 14 – Network Security

Volume 15 – MPLS-TP

Volume 16 – Ultra Low Latency (ULL) Testing

Volume 17 – Impairments

Volume 18 – LTE Access

Volume 19 – 802.11ac Wi-Fi Benchmarking

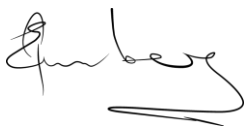
Volume 20 – SDN/OpenFlow

Volume 21 – Network Convergence Testing

Volume 22 – Testing Contact Centers

A soft copy of each of the chapters of the books and the associated test configurations are available on Ixia's Black Book website at http://www.ixiacom.com/blackbook. Registration is required to access this section of the Web site.

At Ixia, we know that the networking industry is constantly moving; we aim to be your technology partner through these ebbs and flows. We hope this Black Book series provides valuable insight into the evolution of our industry as it applies to test and measurement. Keep testing hard.



Errol Ginsberg, Acting CEO

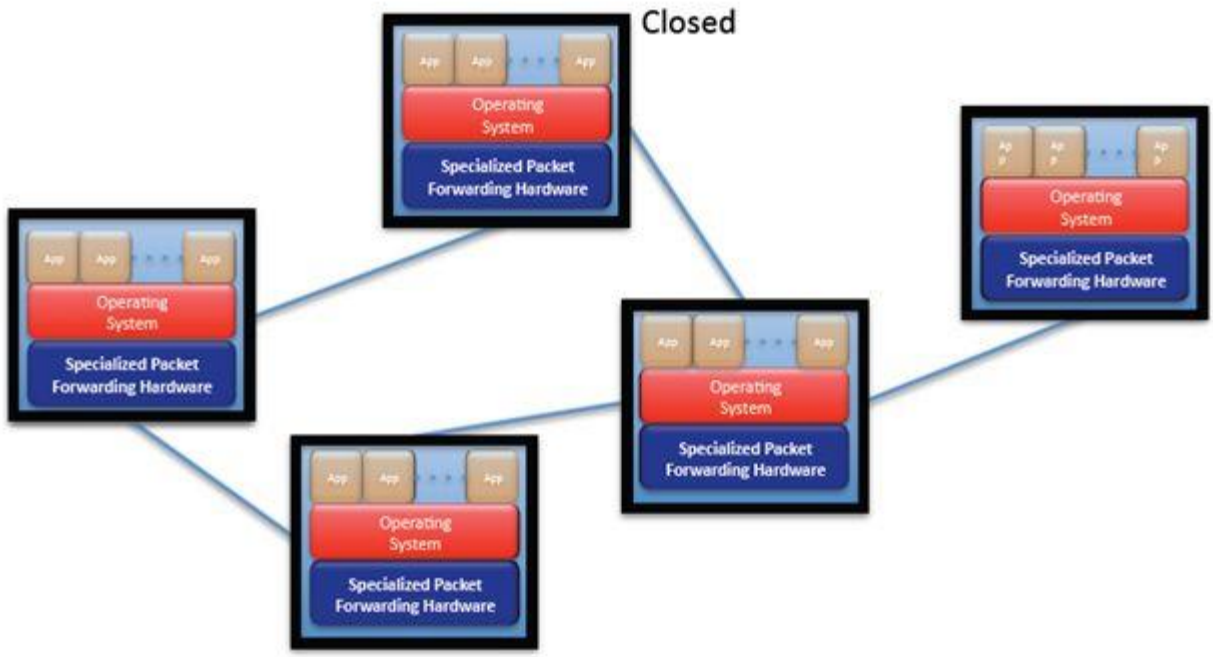
SDN/OpenFlow
Test Methodologies

The tests in this booklet describe detailed methodologies to verify the functionalities and performance of SDN implementations including OpenFlow.

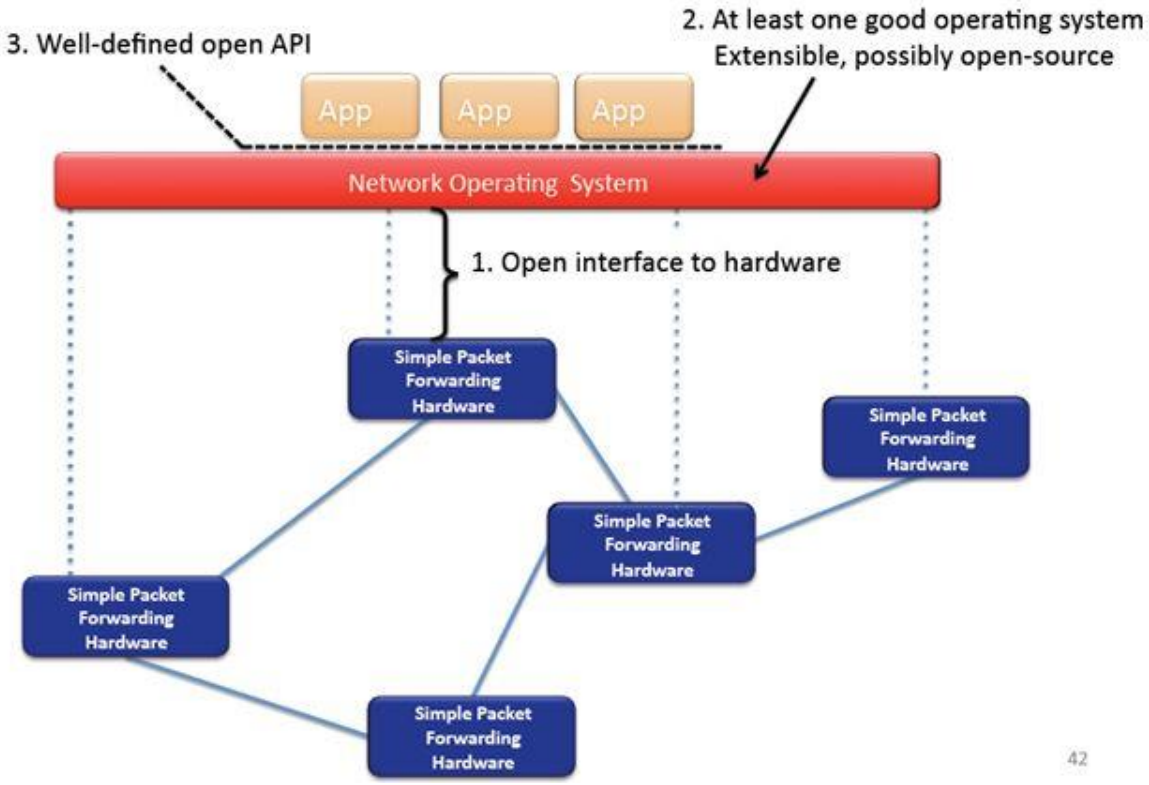
Introduction – Software Defined Networking

Most modern day network architectures rely on a traditional and conventional hierarchical organization, dependent on a tree-like structure of Ethernet switches and routers. Focusing solely on client-server computing, the network architectures fail to meet the needs of today's computing trends. With the changes in traffic patterns for increased accessibility and connectivity, the rising prominence of both private and public cloud services, and the immense parallel server processing necessary for mega datasets, it is imperative that the demand for higher network capacity is fulfilled.

By using software defined networking (SDN), it becomes possible to address these needs using a more dynamic and flexible networking architecture. SDN moves away from traditional architecture and to a revolutionary service delivery platform that can easily and readily address the changes in industry. With SDN, the control plane is accessed and modified using open protocols through software clients. By allowing third parties increased access to the control plane via software, SDN provides enterprises and carriers unparalleled programmability and network flexibility with rapid experimentation and optimization to address business needs.



Current Networking Systems



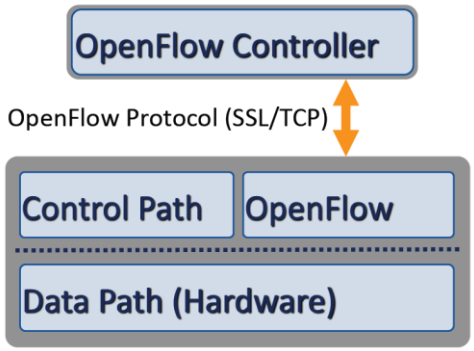
Software Defined Networking System
(Images source: Open Networking Foundation)

OpenFlow

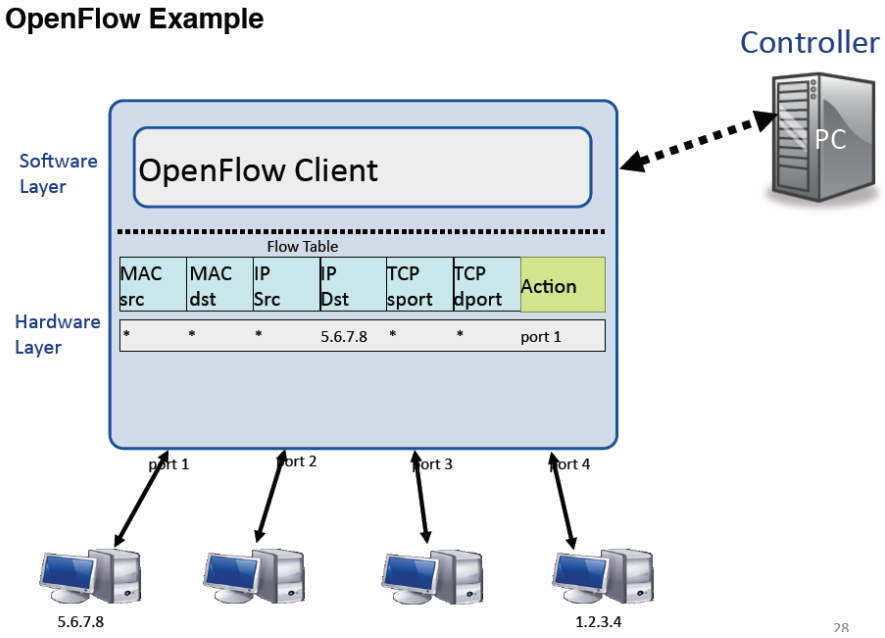
OpenFlow is one such communication protocol that enables SDN. OpenFlow, the first standard interface communications protocol designed specifically for SDN, decouples the control- and data-planes so that software can determine the network packets passing through a network thereby customizing the needs of applications and its users. With the centralization of the control plane, it is possible to introduce and experiment with new capabilities in isolated slices of the network without affecting the rest of the network. This major change in network architecture offers its users a way to introduce new applications without the reliance upon individual device configuration and vendor releases.
SDN via OpenFlow revolutionizes and expands the capabilities of networking architecture, providing key benefits for the ever-changing market. With rapid innovation and experimentation possible through software control, OpenFlow delivers the flexibility necessary to combat current and future network problems. Additionally, not only is there an increased choice regarding new applications but there is also an increased choice regarding vendor markets. The switch from a hardware-based to a software-based networking architecture creates open multivendor markets as the network operator can select different control- and data-plane vendors. The division of the planes increases network reliability and security, creating the potential to lower both CAPEX and OPEX costs while decreasing the complexity of networking hardware and network management.

OpenFlow Basic

OpenFlow defines two main device types; a controller and a switch. The OpenFlow controller talks to each OpenFlow switch over an IP connection (known as OF Channel) and has the ability to program the forwarding table of the switch with flow-table entries.

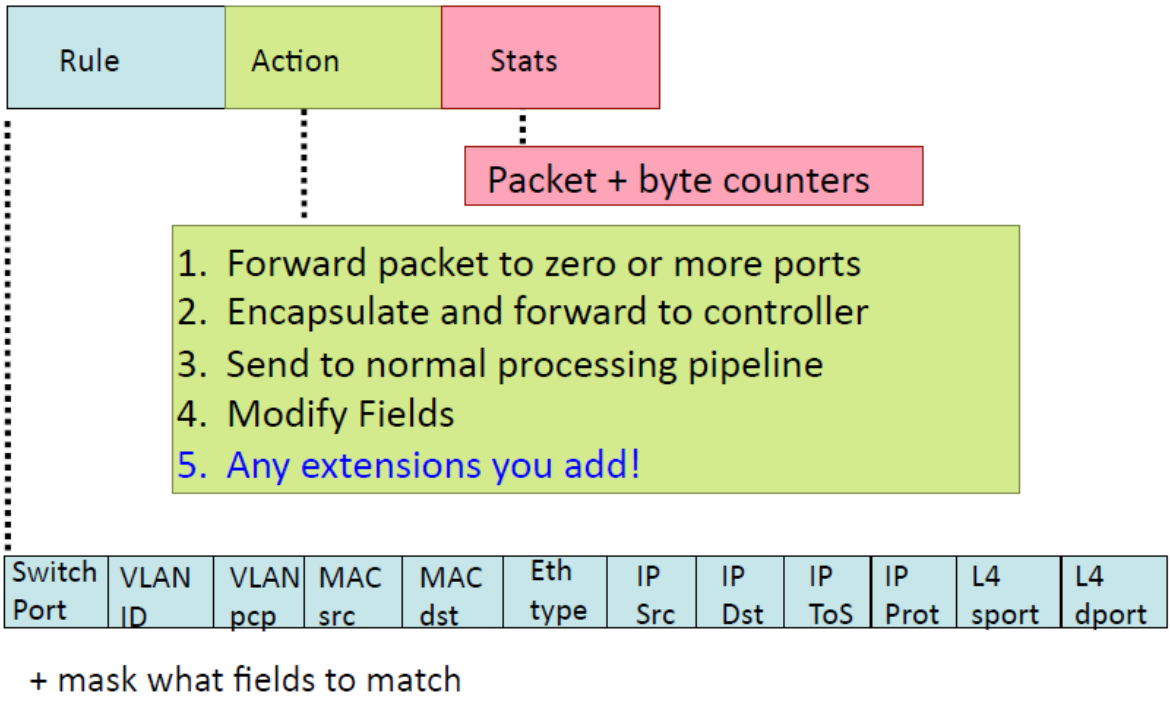


These Flow Table entries are called Flows. A Flow has set of match fields and related actions. A match fields define the packet match criteria for the switch. Match fields are various protocol fields such as L2 MAC address, L3 IP address, VLAN address, etc. For each set of Match, there is a corresponding Action associated with it. The action defines what the switch supposed to do when packets matches the Match criteria. An Action could set certain protocol fields such as VLAN address and/or forward the packet to a port. A port could be a physical port or it could be virtual port number to identify an operation such as flood.

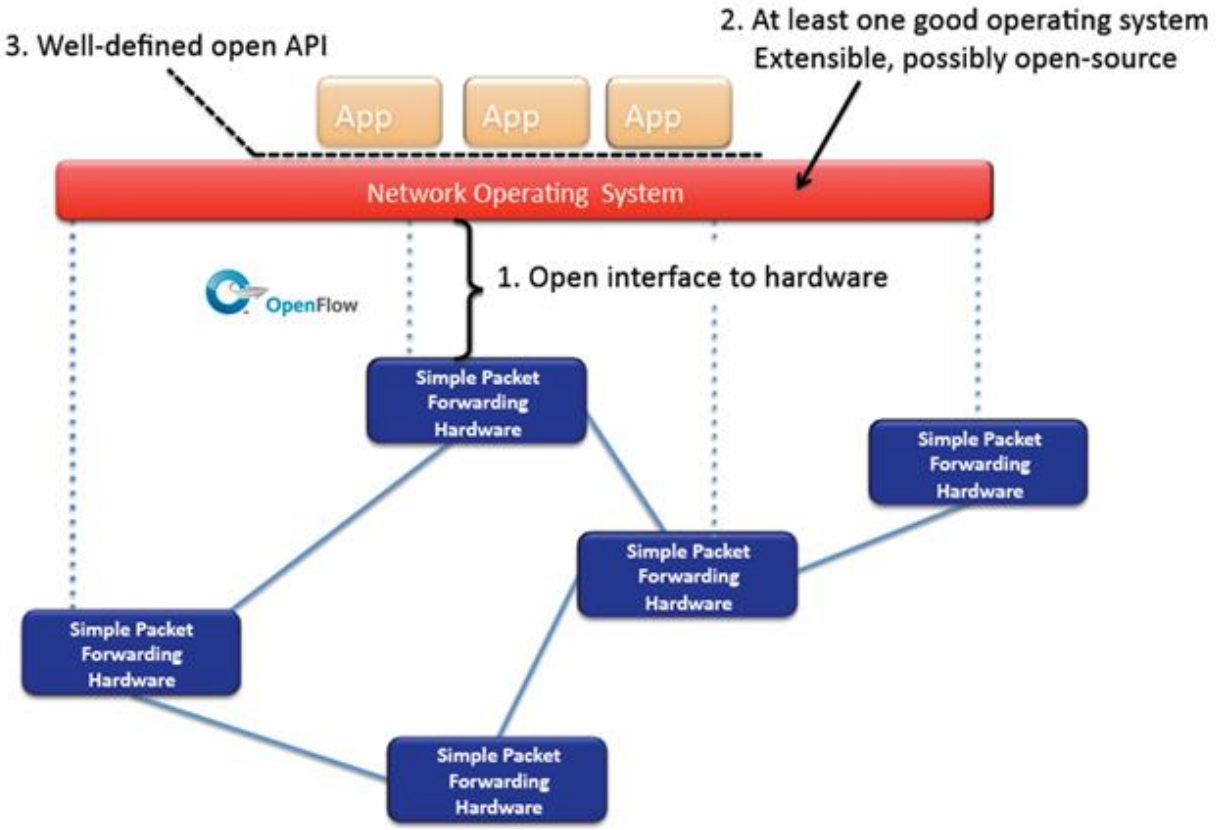


When a packet enters a switch, the switch performs match criteria on the packet by looking up its Flow Table. When a packet matches a Flow table entry, switch performs a corresponding match associated with that flow entry. Please note that not all the match fields need to be

defined. A wildcard is used to match for all the values for a certain match field. Along with match and action, a flow entry also has stats counter. The counters indicated the packet count for each flows.



The SDN applications run on top of the OpenFlow controller using a well-defined API, known as the north-bound API. Each controller vendor provides their own set of APIs. Applications can range from layer 2 or layer 3 learning networks to static provisioned networks and can be as simple or complicated as the requirements demand. New applications are being developed every day to address challenges in the data center, service provider WAN, enterprise, and other networks.


3. Well-defined open API
2. At least one good operating system
Extensible, possibly open-source
App
App
App
Network Operating System
OpenFlow
1. Open interface to hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware

Test Case: OpenFlow Switch Setup and Functional Test

Overview

One of the most important aspects of OpenFlow protocol is to create OF Channel. OF Channel establishes connection between the controller and switch using TCP or TLS. After the TCP session is established, controller and switch exchanges an OFPT_Hello message. The version field in the message is set to the highest OpenFlow protocol version supported by the sender. After receiving the message, the recipient calculates the OpenFlow protocol version to be used. The lowest version that is sent and received successfully is used as the OpenFlow protocol version.

After version negotiation, the controller sends Features Request message and switch sends Features Reply message to advertise their capabilities. Then Echo Request and Echo Reply messages are exchanged to keep the OF Channel session alive between the controller and the switch.

Ladder diagram

The following diagram illustrates the message exchange between the switch and the controller.

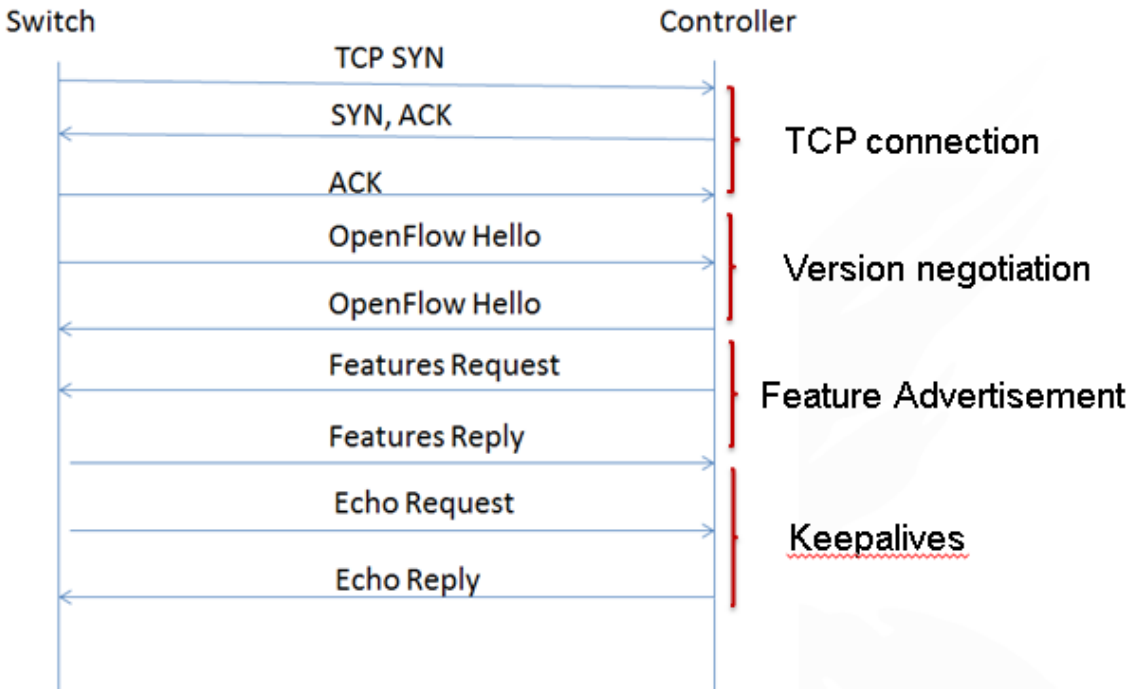


Figure 1: Message Exchange between Switch and Controller

Objective

The OpenFlow Switch Setup and Functional test verifies the functionality of OpenFlow switches. The test provides basic guidelines on how to configure OpenFlow controller, establish OF Channel, and retrieve switch capabilities using learned information. The test also trigger stat request using on demand message function and verifies that switch sends reply with requested information. At the end of the test, statistics are reviewed.

Setup

The following figure illustrates the test setup.

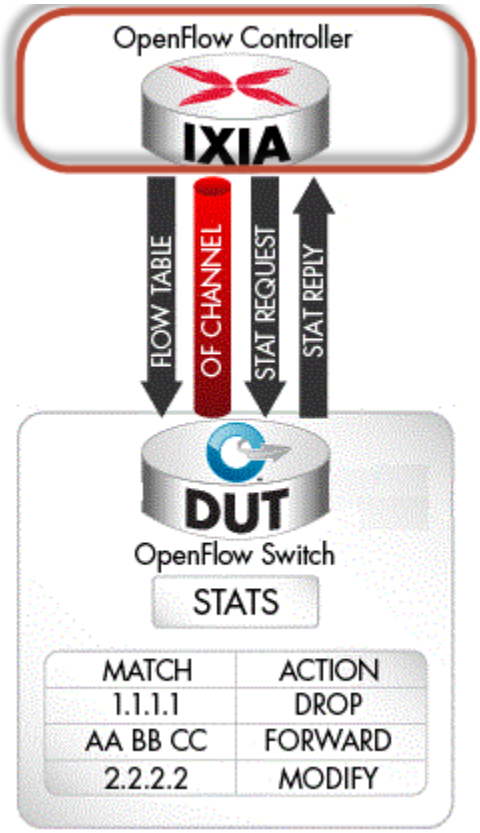


Figure 2: Test Setup – OpenFlow Switch Setup and Functional Test, test setup

Step-by-Step Instructions

The following steps describe the procedure for performing the test.

1. Reserve one Ixia port

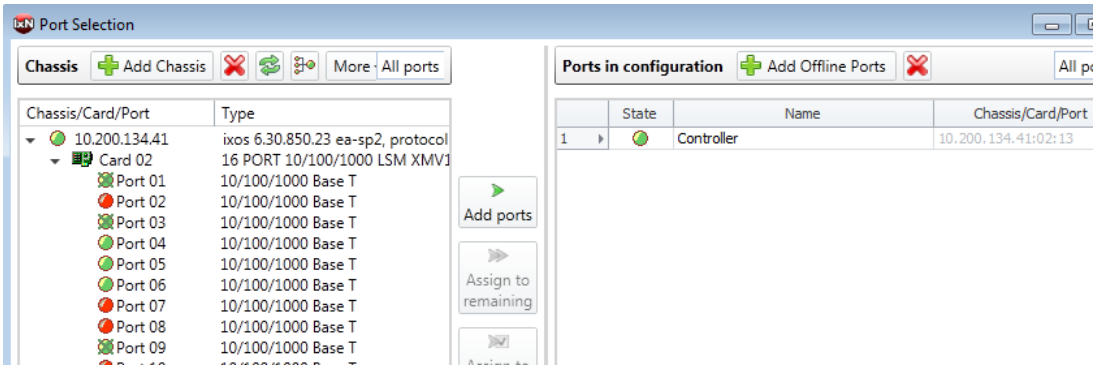


Figure 3: Port Selection window

2. Enable OpenFlow by selecting the **OpenFlow** check box in the **RoutingSwitching** tab in the **Protocols** window.

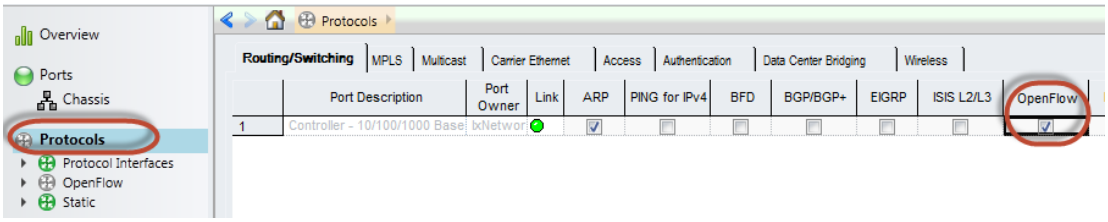


Figure 4: Routing/Switching tab, Protocols window

3. Configure the emulated controller IP address and **Gateway** address from the **Connected Interface** tab on the **Protocol Interfaces** window. Use the IP address of the OpenFlow switch if you have only one switch. For Of Channel, ensure that ARP is resolved.

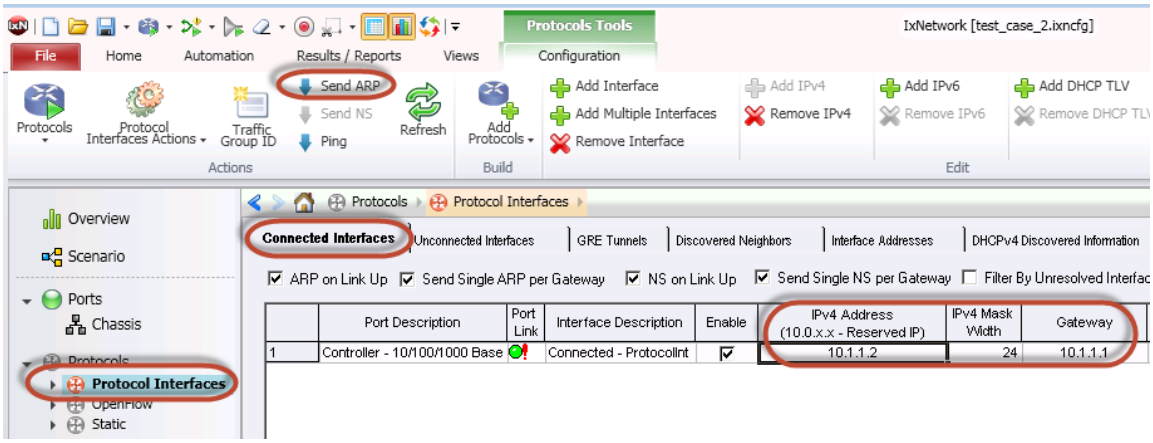


Figure 5: Protocol Interfaces window

4. Define the port role by selecting the role from the **Port Role** list on the **Ports** tab on the **OpenFlow** window. You can select any of the following port roles:
    a. Control: Ixia port will only act as a Controller.
    b. Traffic: Ixia port will be used as traffic endpoints.
    c. Control & In-Band Traffic: Ixia port will act as emulated controller as well as traffic endpoints (that is, in-band signaling).

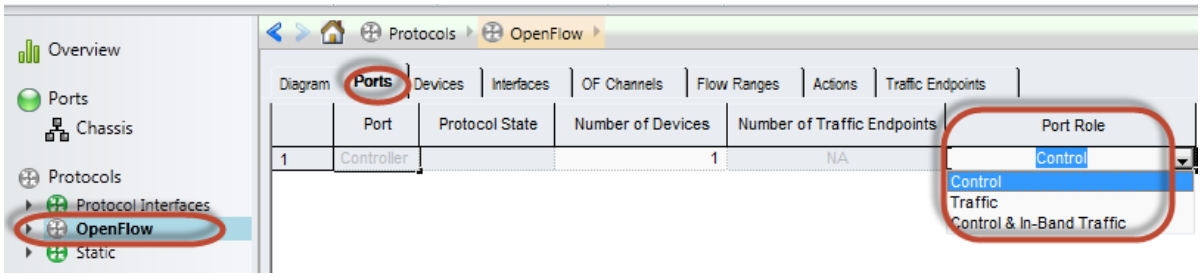


Figure 6: Ports tab, OpenFlow window

5. Configure the **Number of Interfaces** by going to the **Devices** tab on the **OpenFlow** window. The number of interfaces should be equal to the number of emulated NICs of a controller.

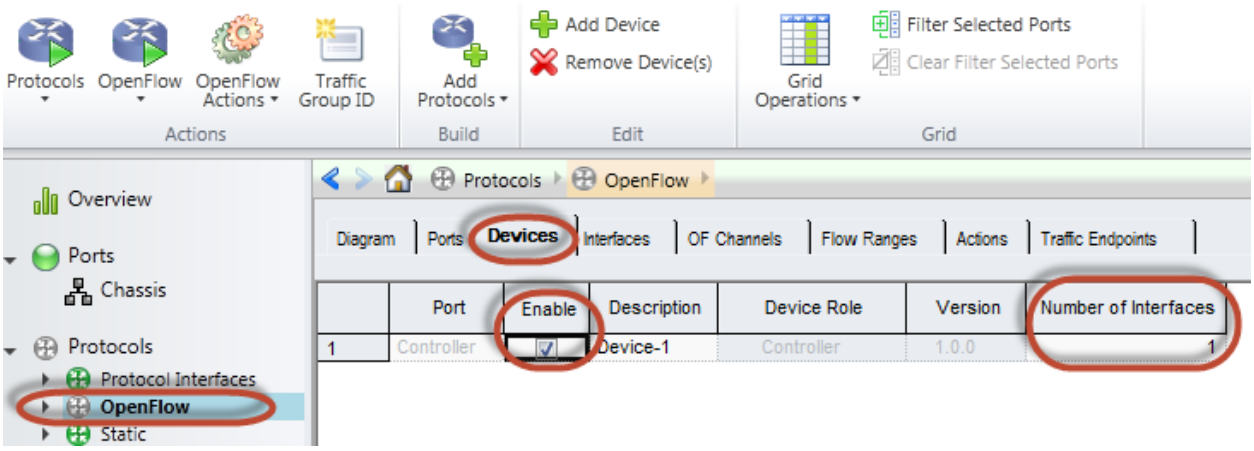


Figure 7: Devices tab, OpenFlow window

6. Go to the **Interface** tab of the **OpenFlow** window and assign the **Protocol Interfaces** that you created on the **Protocol Interface** window. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1*.

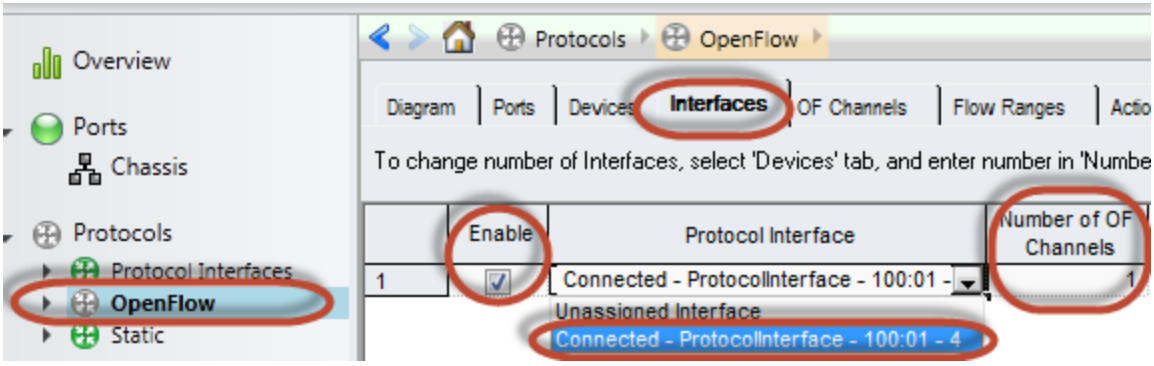


Figure 8: Interfaces tab, OpenFlow window

Following are some of the important parameters available on the **Interface** tab of the **OpenFlow** window:

- **Periodic Echo**: Used to keep OF Channel session alive
- **Mode of Connection**: Used to indicate whether Ixia port initiates TCP connection. The available options are **Passive** and **Active**. **Passive** is selected by default and it indicates that the Ixia port will not initiate the TCP connection. **Active** indicates that the Ixia port will be used to initiate the TCP connection.
- **TCP Port**: Indicates the port is used to setup OF Channel. The default is 6633.

- **Delete all Flow at Startup**: Used to ensure that the switch does not have any pre-installed flow in its table. If this check box is selected, Ixia emulated controller will send Flow Delete message with all 12 Tuples set as wildcard (*) after the OF Channel is up. And the test starts with no pre-installed flows.

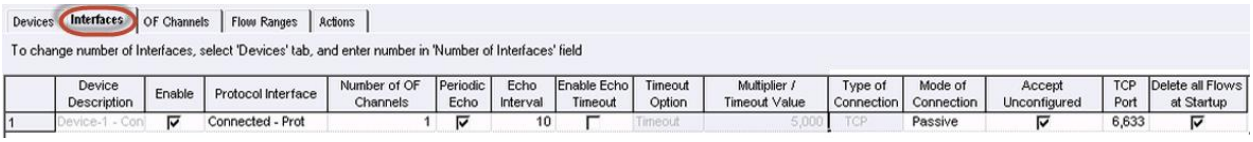


Figure 9: Interface tab, OpenFlow window parameters

7. Go to the **OF Channels** tab on the **Controller** window and enable OF Channel by selecting the **Enable** check box. Also, enter the DUT IP address in the **Remote IP** field. The IP address that you enter in the Remote IP field is the IP address of the OpenFlow switch.

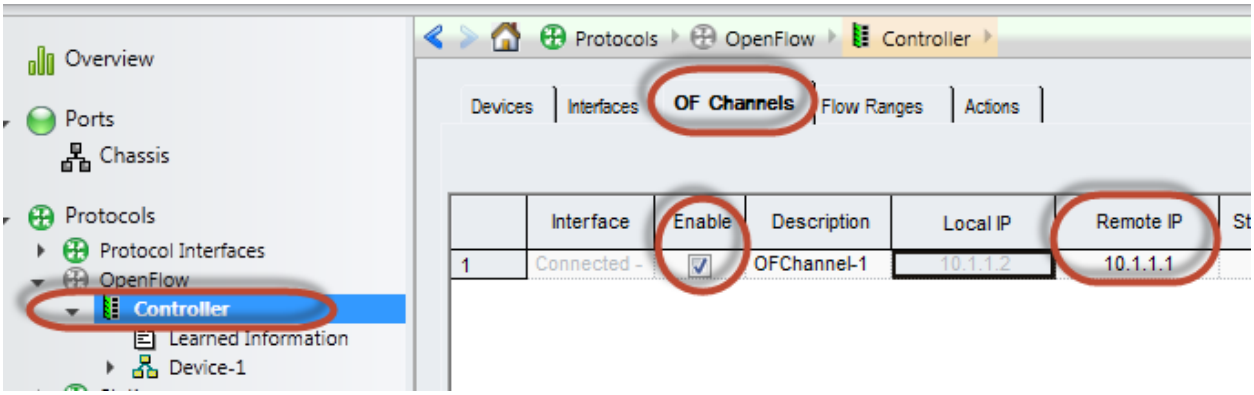


Figure 10: OF Channels tab, Controller window

8. Enable control capture by selecting the **Control Enable** check box on the **Captures** window. Also, start capture using the **Capture** control on the ribbon.

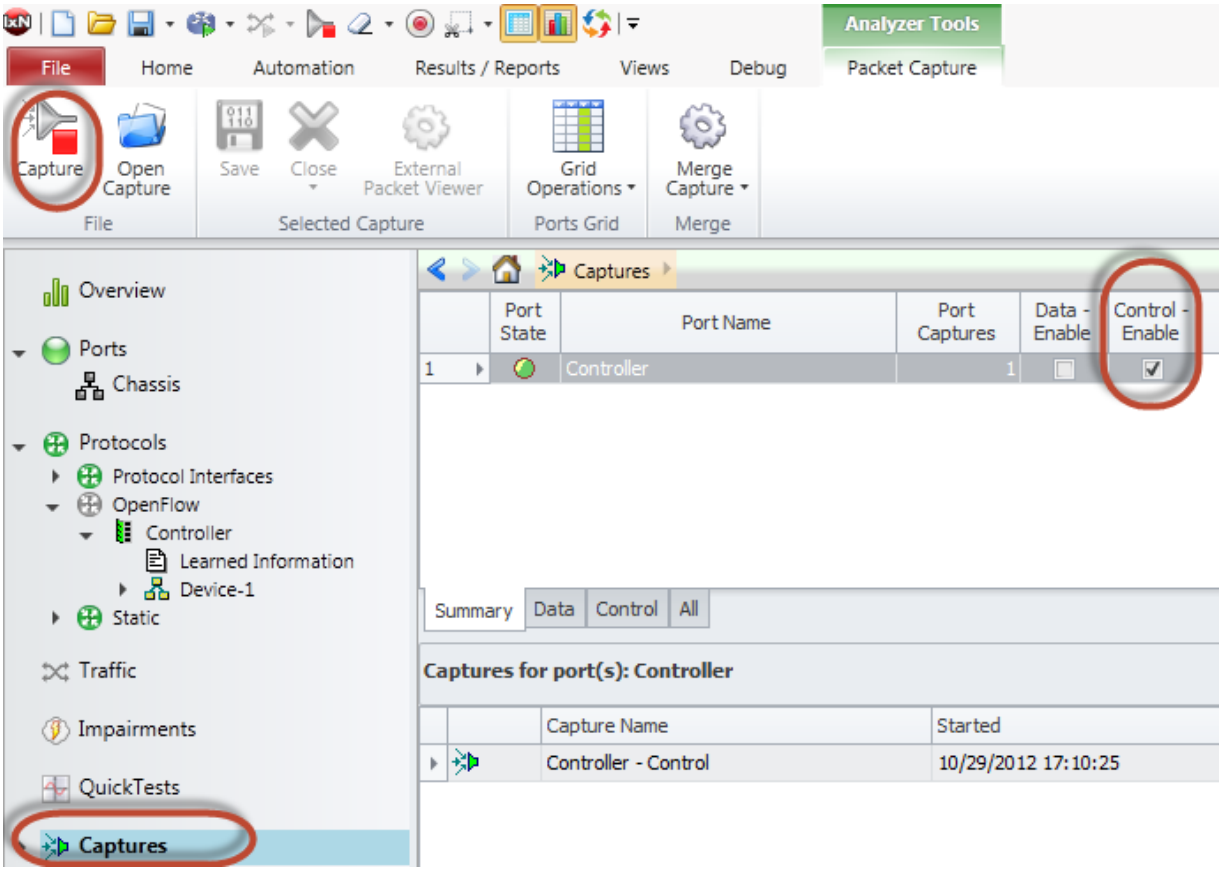


Figure 11: Captures window

9. Start OpenFlow protocol from the **Protocols** control on the ribbon.

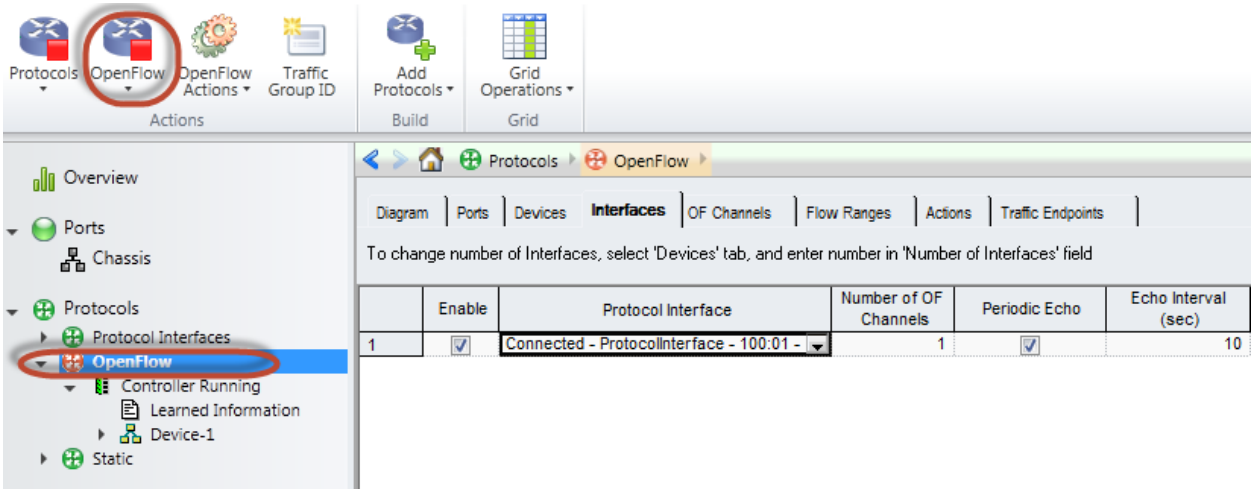
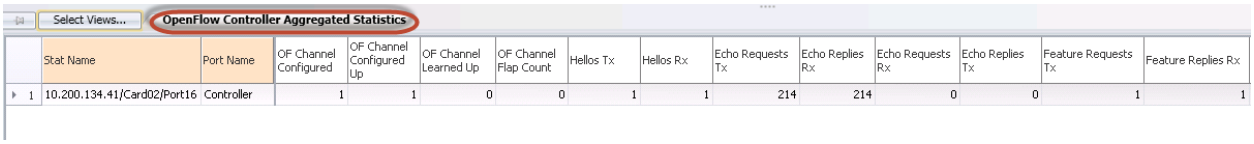


Figure 12: OpenFlow window

Results Analysis

The test results are available on the **OpenFlow Controller Aggregated Statistics** tab.



	Stat Name	Port Name	OF Channel Configured	OF Channel Configured Up	OF Channel Learned Up	OF Channel Flap Count	Hellos Tx	Hellos Rx	Echo Requests Tx	Echo Replies Rx	Echo Requests Rx	Echo Replies Tx	Feature Requests Tx	Feature Replies Rx
1	10.200.134.41/Card02/Port16	Controller	1	1	0	0	1	1	214	214	0	0	1	1

Figure 13: OpenFlow Controller Aggregated Statistics tab

This Statistics tab shows detailed information on OpenFlow connection status, message exchange, error condition, and packet_ins.

You can verify the following statistics to analyze the OF Channel connection:

OF Channel Configured	Displays number configured OF Channel
OF Channel Configured UP	This statistics displays status of the configured OF Channel
OF Channel Learned UP	By default Ixia emulated controller accepts OF Channel connection from a switch even if it is not configured. This statistics shows the un-configured OF Channel. Note: The Configure OF Channels option under Learned Information allows configuring the learned OF Channel.
OF Channel flap count	This statistics shows the number of times the TCP session is reset.
Hello Tx/Rx	This statistics displays hello message exchange.
Echo request Tx/Rx	This statistics displays echo message for the liveliness between the switch and controller.

If the **OpenFlow Controller Aggregated Statistics** tab is not available, you need to enable it from the **Select Views** window.

To enable the **OpenFlow Controller Aggregated** Statistics tab, click **Select Views** and select **OpenFlow Controller Aggregated Statistics** check box as shown in the following figure.

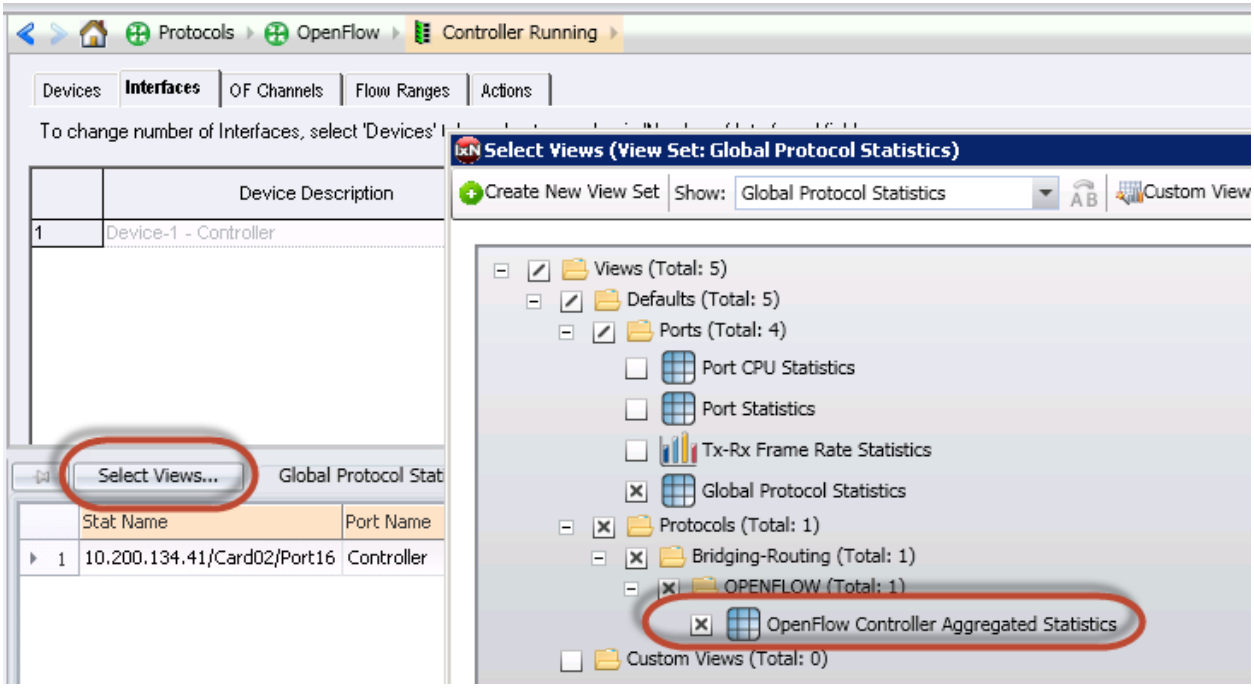


Figure 14: Select Views window

Go to **Capture Analyzer** and click on **Ladder Diagram** to verify message exchange between the controller and the switch.

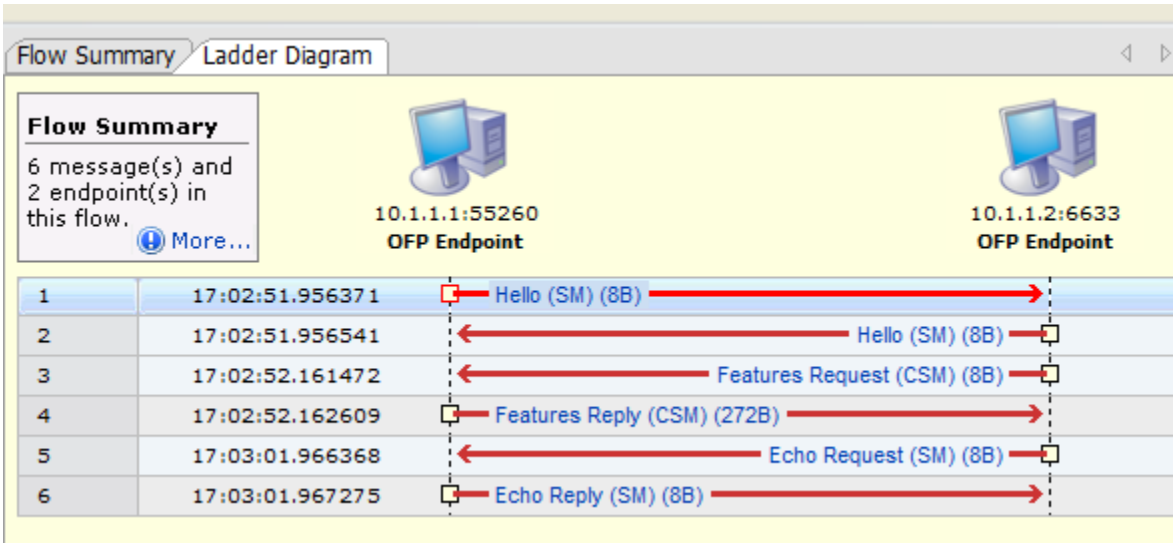


Figure 15: Capture Analyzer, Ladder Diagram tab

To verify the switch capabilities, supported action or any error condition, go to **Learned Information** window. The Learned Information window contains several tabs as shown in the figure below.. Click **Refresh** button on the ribbon to update this information.

The **OF Channel Learned Info** tab, contains multiple panes. Left pane displays OF Channel information including **TCP port**, **Data Path ID**, **Reply State** and any error message received from the switch. When you select a row (OF Channel), the right pane displays all OpenFlow enabled ports information on that switch.

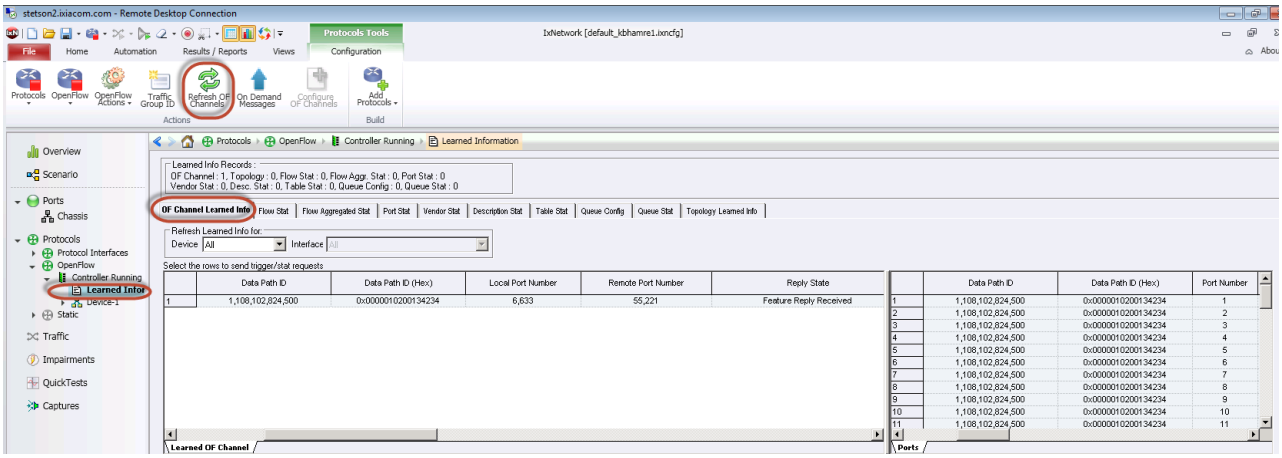


Figure 16: OF Channel Learned Info tab, Learned Info window

The **Ports Stat** view displays the details of each OpenFlow ports of the connected switch.

Various OpenFlow message can be sent from controller using On Demand Message button on the ribbon. Select the OF Channel and then click On Demand Message button. On the OpenFlow Learned Info Trigger Settings window, select the Port Stat check box (Multiple stats request can be sent) and click OK.

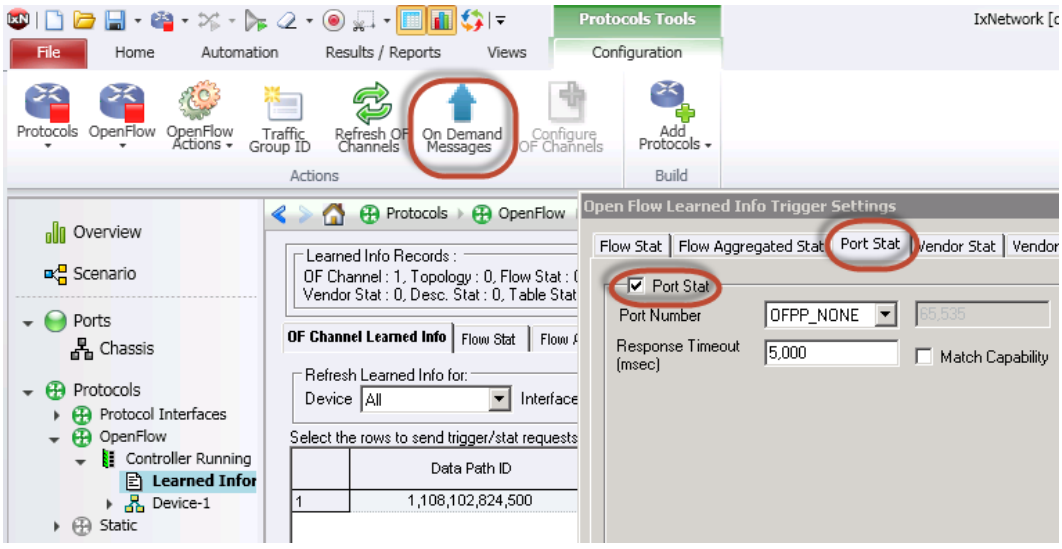
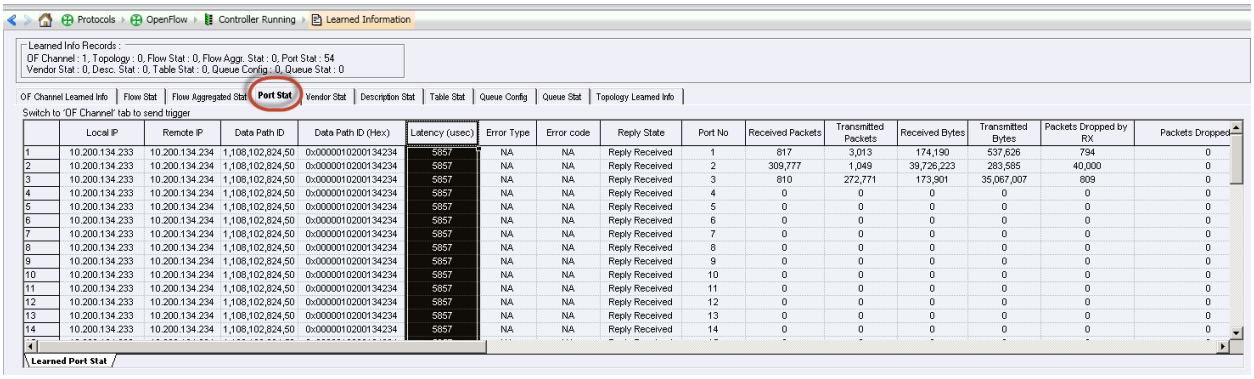


Figure 17: OpenFlow Learned Info Trigger Settings window

This feature allows you to monitor the switch status without logging on to the switch. In this example, you can view OpenFlow switch ports information like Tx and Rx packet counts, dropped packet, CRC, Frame alignment, Collision and Overrun errors. The Latency field shows the response time of the switch for the particular stat request.



	Local IP	Remote IP	Data Path ID	Data Path ID (Hex)	Latency (usec)	Error Type	Error code	Reply State	Port No	Received Packets	Transmitted Packets	Received Bytes	Transmitted Bytes	Packets Dropped by RX	Packets Dropped
1	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	1	817	3,013	174,190	537,626	794	0
2	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	2	309,777	1,049	39,726,223	283,585	40,000	0
3	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	3	810	272,771	173,901	35,067,007	809	0
4	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	4	0	0	0	0	0	0
5	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	5	0	0	0	0	0	0
6	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	6	0	0	0	0	0	0
7	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	7	0	0	0	0	0	0
8	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	8	0	0	0	0	0	0
9	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	9	0	0	0	0	0	0
10	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	10	0	0	0	0	0	0
11	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	11	0	0	0	0	0	0
12	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	12	0	0	0	0	0	0
13	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	13	0	0	0	0	0	0
14	10.200.134.233	10.200.134.234	1,108,102,824,50	0x0000010200134234	5857	NA	NA	Reply Received	14	0	0	0	0	0	0

Figure 18: Port Stats tab showing OpenFlow switch ports information

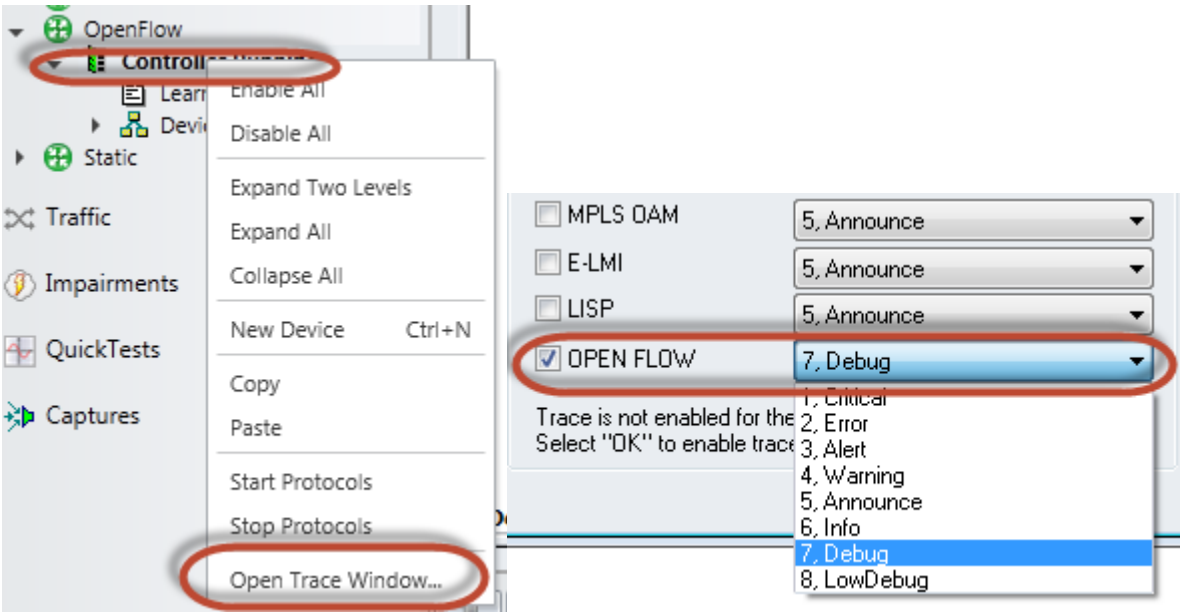
Troubleshooting

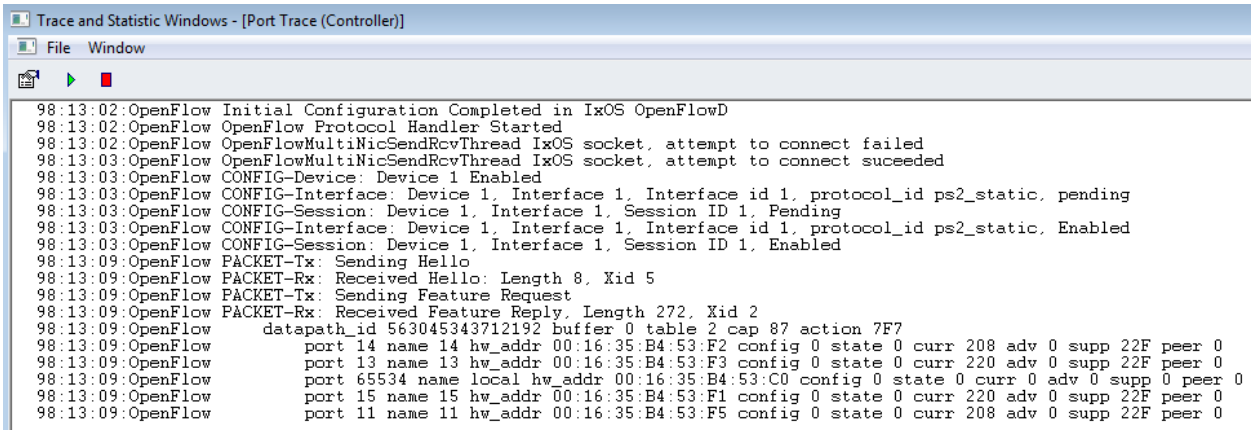
Use following steps to troubleshoot any OF Channel issues:

- Enure **ARP** is resolved under **Protocol Interface**. Also try to PING emulated controller IP (Ixia) from OpenFlow switch.
- You can enable protocol trace from the trace window.

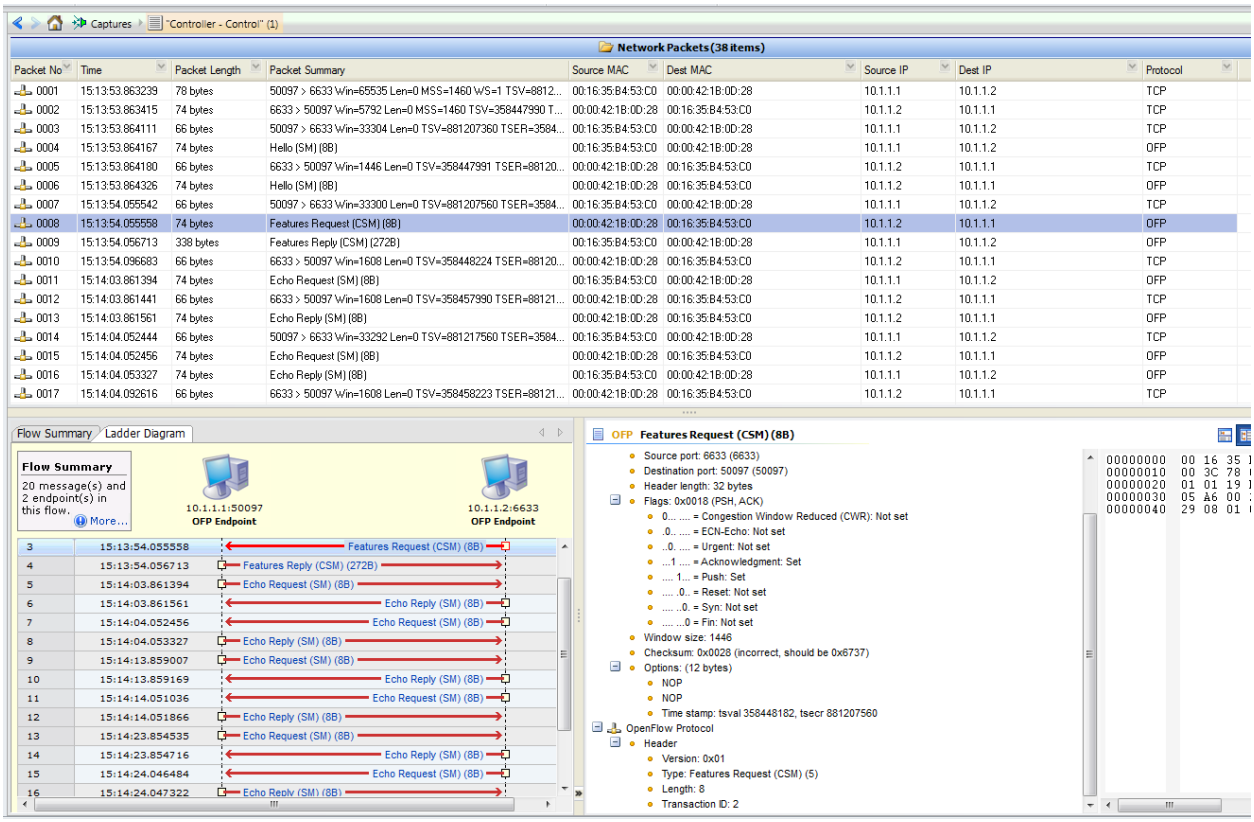
To open the trace window, right-click on **Controller Interface** and then click **Open Trace Window**.

On the trace window select the **OPEN FLOW** check box and then select the debug level from the list and click **OK**.





IxNetwork Analyzer can decode OpenFlow messages. Use **Control**-plane capture to see **bi-directional** communication in real-time (Note - It requires Analyzer Chassis component license). Using this trace, you can determine whether bi-directional communication is happening properly as per ladder diagram shown earlier.



Conclusions

By validating the statistics and control-plane message exchanges using the features above, we have verified that the DUT can successfully establish OF Channel, keep the session alive and respond to Stats Request from the controller.

Test Case: OpenFlow Switch Forwarding Test

Overview

Through OpenFlow you can program data path by building the flow table in OpenFlow switch. In the flow table there are two components: Match and One or more Actions.

OpenFlow 1.0 specification covers 12 tuple matches as shown below.

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

After the match, certain actions can be performed, such as forward packet to zero or more ports, modify the field, drop the packet or if no match found forward it to controller.

Objective

The objective of this test is to verify the ability of the OpenFlow switch to forward L2 traffic. The DUT should be able to look up the Flow Table when L2 traffic is received and forward the traffic based on specified actions. In this test, initially controller will push down L2 flows with certain Match and Action parameters. Then using traffic wizard, matching traffic will be created and sent. Using the traffic statistics; switch forwarding performance will be verified.

Setup

The following figure illustrates the test setup:

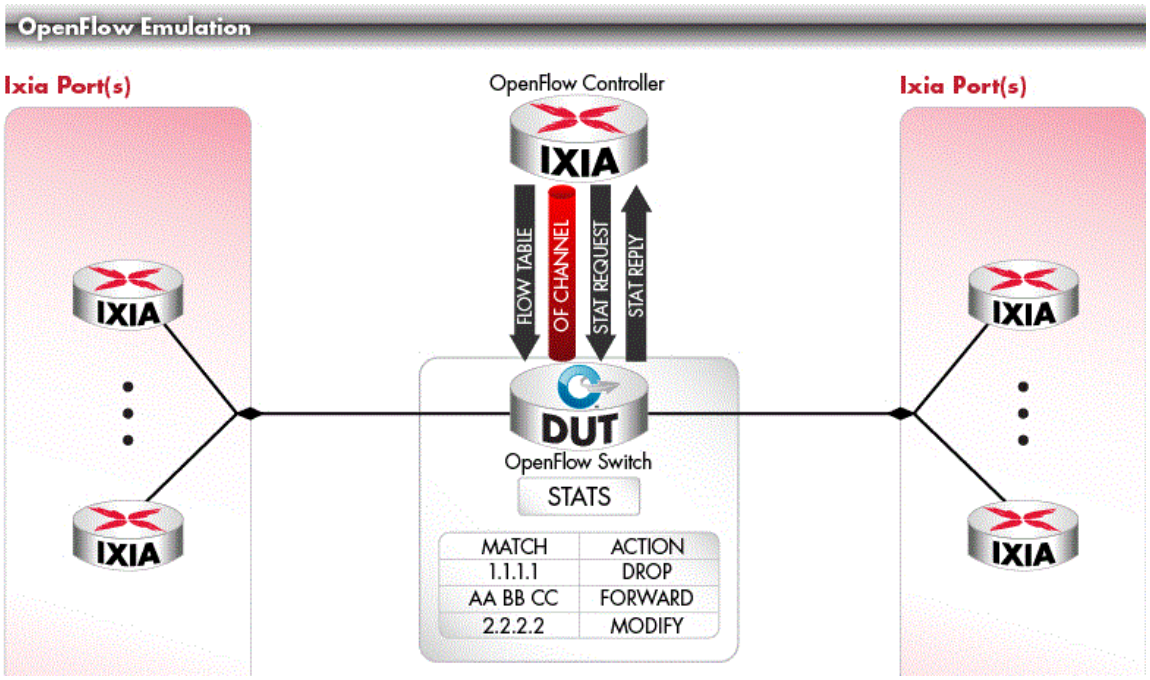


Figure 19: OpenFlow Switch Forwarding Test, test setup

Step-by-Step Instructions

The following steps describe the procedure for performing the test.

1. Reserve 3 Ixia ports (1 for controller and 2 port traffic endpoints)

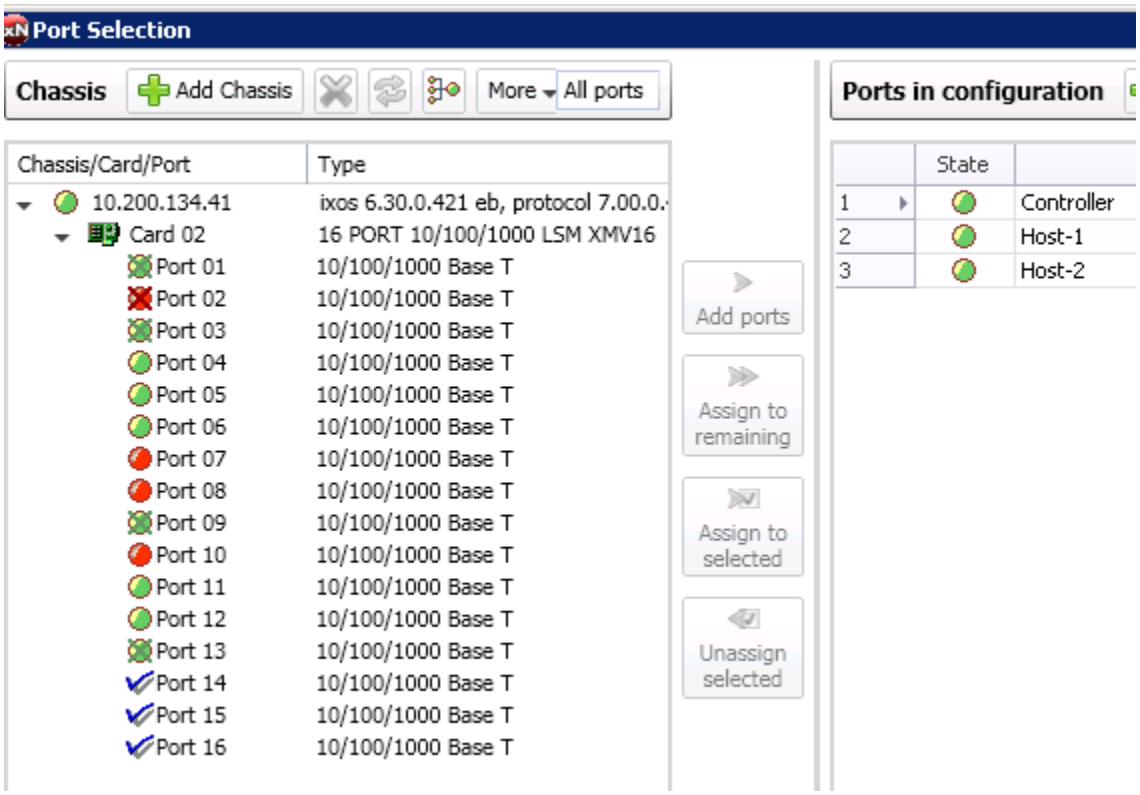


Figure 20: Port Selection window

2. Enable OpenFlow on all the ports by selecting the **OpenFlow** check box on the **Protocols** window.

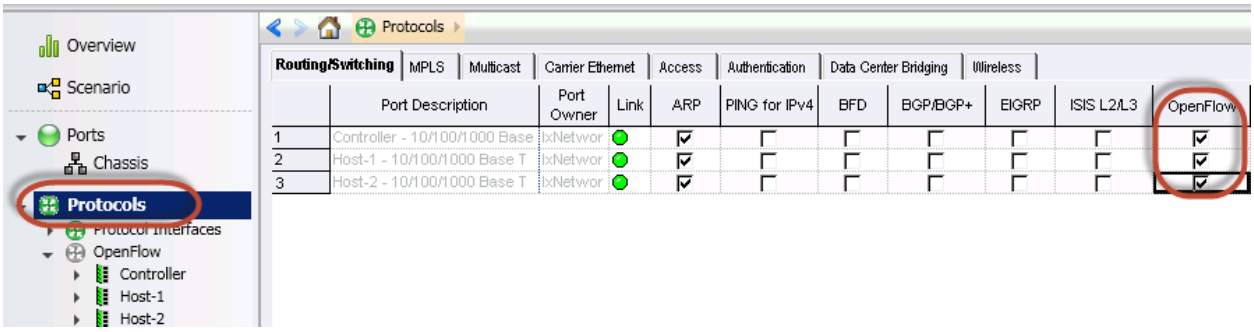


Figure 21: RoutingSwitching tab, Protocols window

3. Configure the emulated controller IP address and **Gateway** address from the **Connected Interface** tab on the **Protocol Interfaces** window. Use the IP address of the OpenFlow switch if you have only one switch. For Of Channel, ensure that ARP is resolved. You do not have to configure anything on host port.

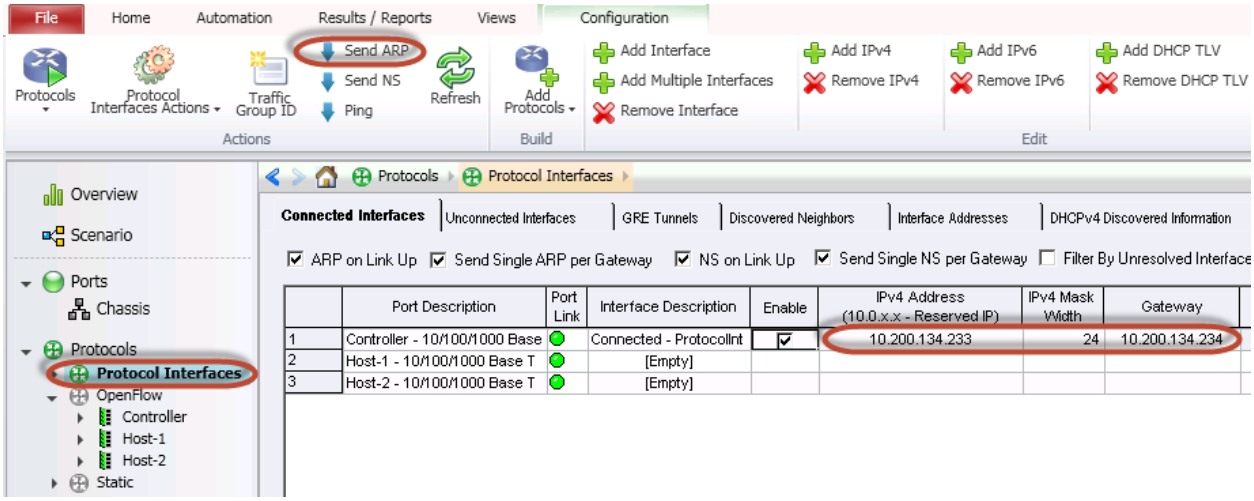
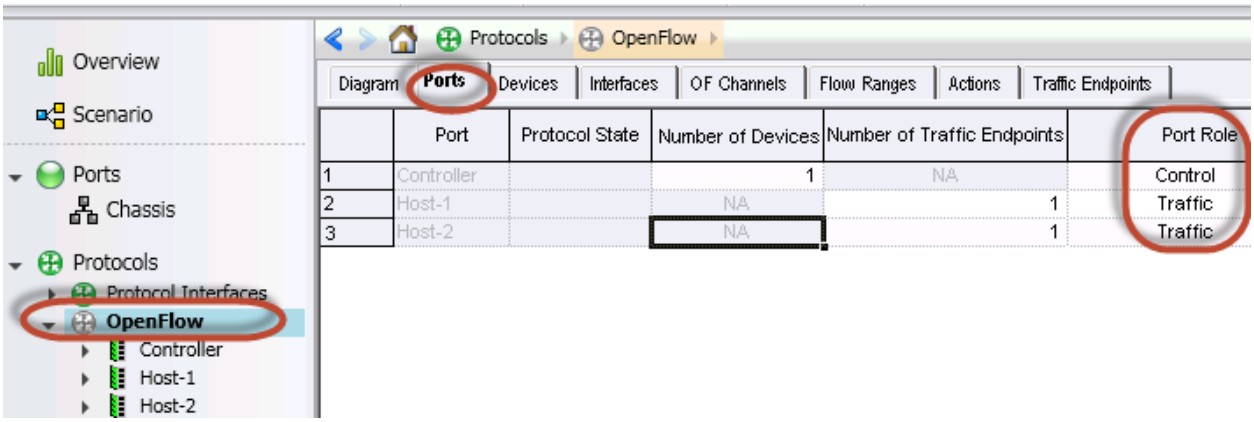


Figure 22: Connected Interfaces tab, Protocol Interface window

4. Define the port role by selecting the role from the **Port Role** list on the **Ports** tab on the **OpenFlow** window. You can select any of the following port roles:
   a. **Control**: for Controller port
   b. **Traffic**: for host ports



5. Configure the **Number of Interfaces** as *1*, by going to the **Devices** tab on the **OpenFlow** window. The number of interfaces should be equal to the number of emulated NICs of a controller.

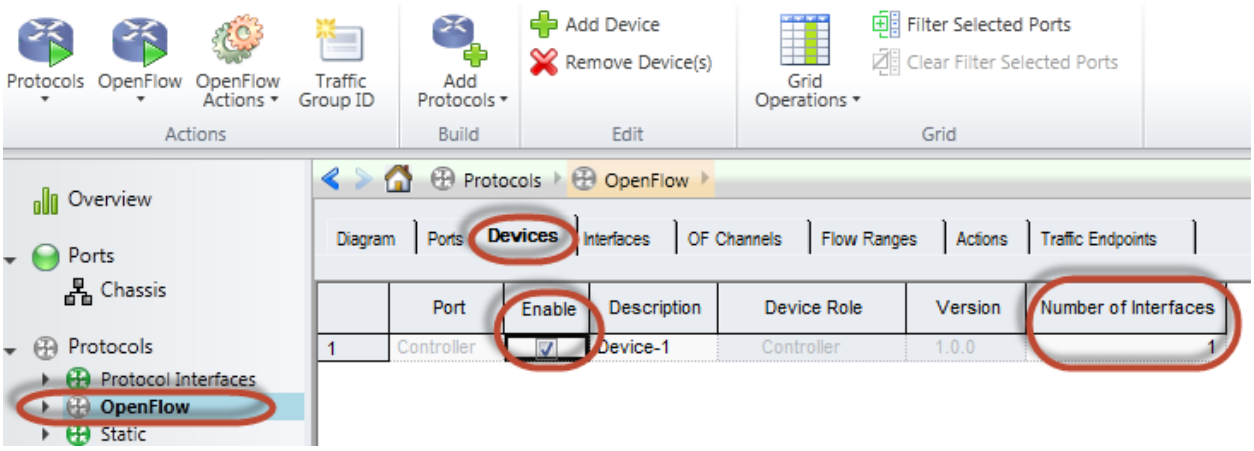


Figure 23: Device tab, OpenFlow window

6. Go to the **Interface** tab of the **OpenFlow** window and assign the **Protocol Interfaces** that you created on the **Protocol Interface** window. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1*.

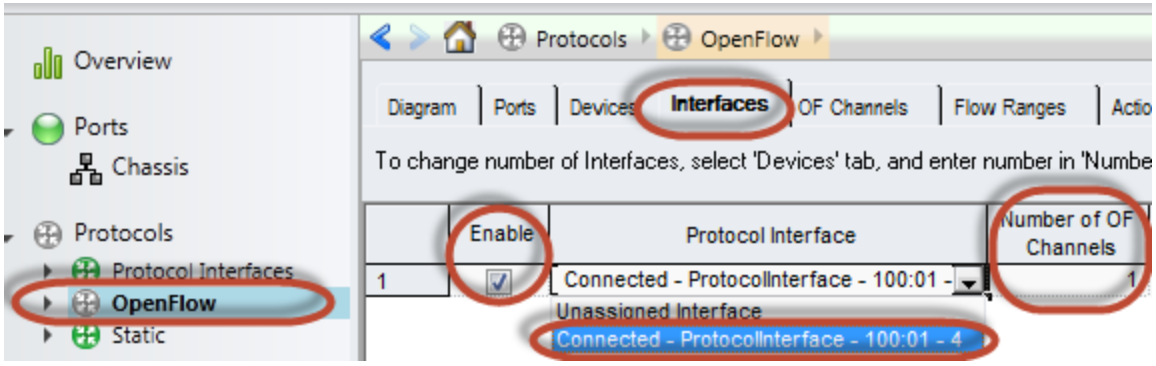


Figure 24: Interfaces tab, OpenFlow window

7. Go to **OF Channels** tab and enter DUT IP address in **Remote IP** field. Change flow range count to *2*.

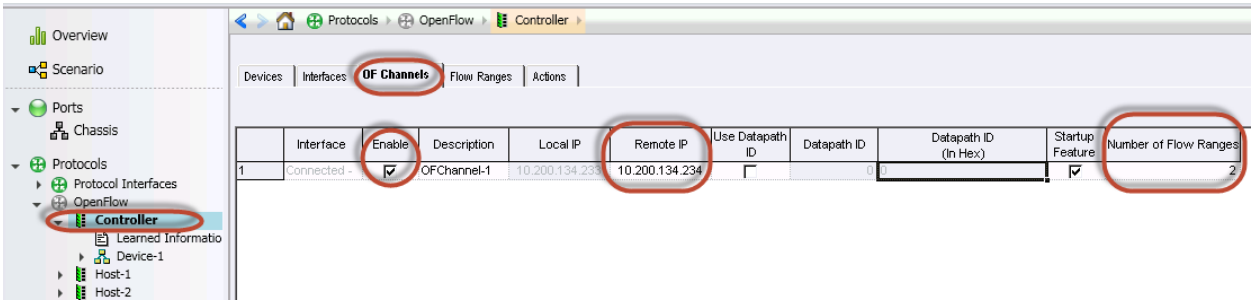


Figure 25: OF Channel tab, OpenFlow window

8. In the **Flow Range** tab, create 5 flows on each range. Enter **Source/Destination MAC and VLAN-ID**. For remaining all field use wild card value (*). This means, Switch will make forwarding decision based on matching Src/Dst MAC address and VLAN-ID. Configure correct **In Port** field with the switch port number that is connected to Ixia host port generating traffic.

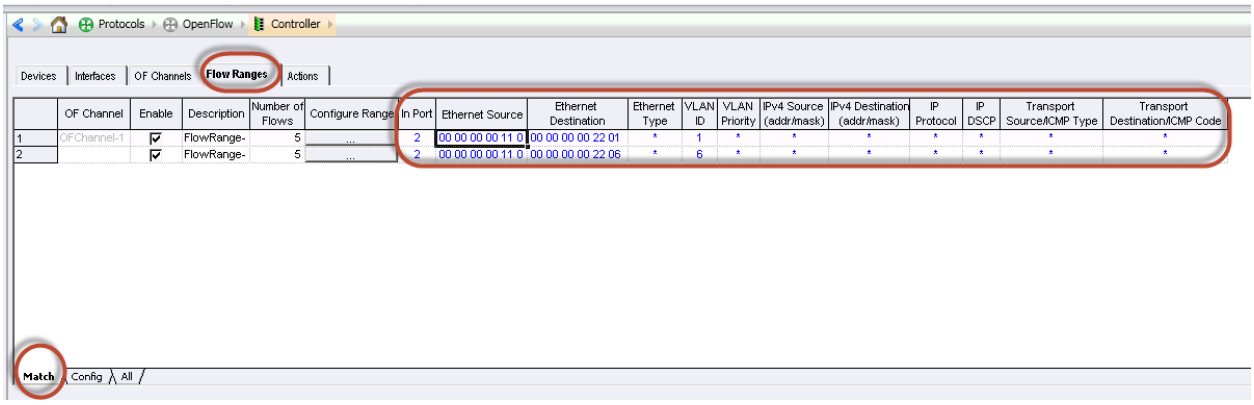
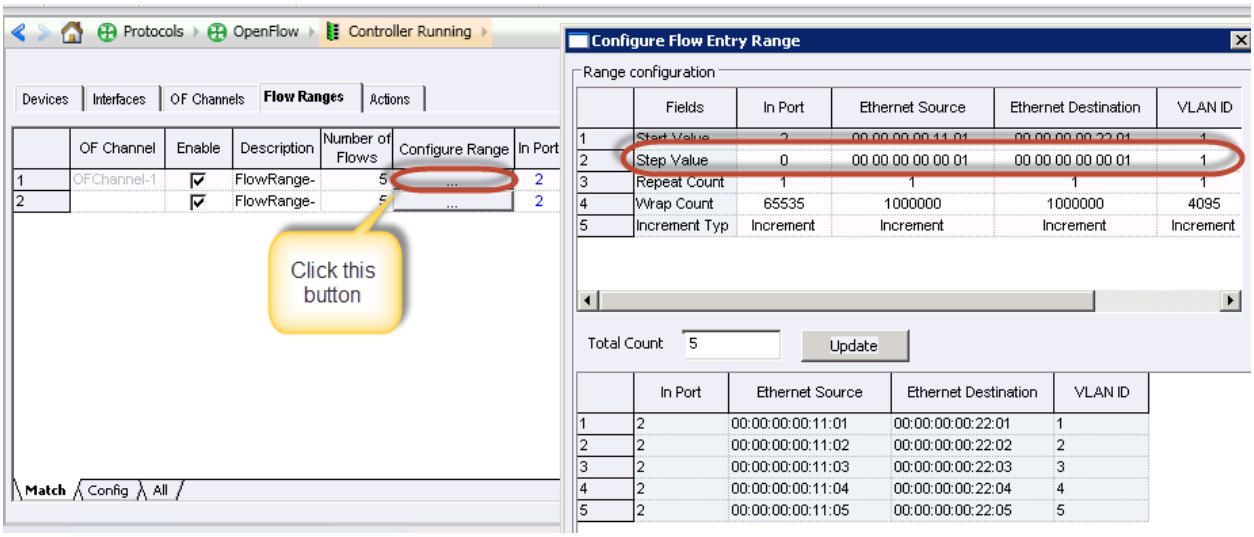
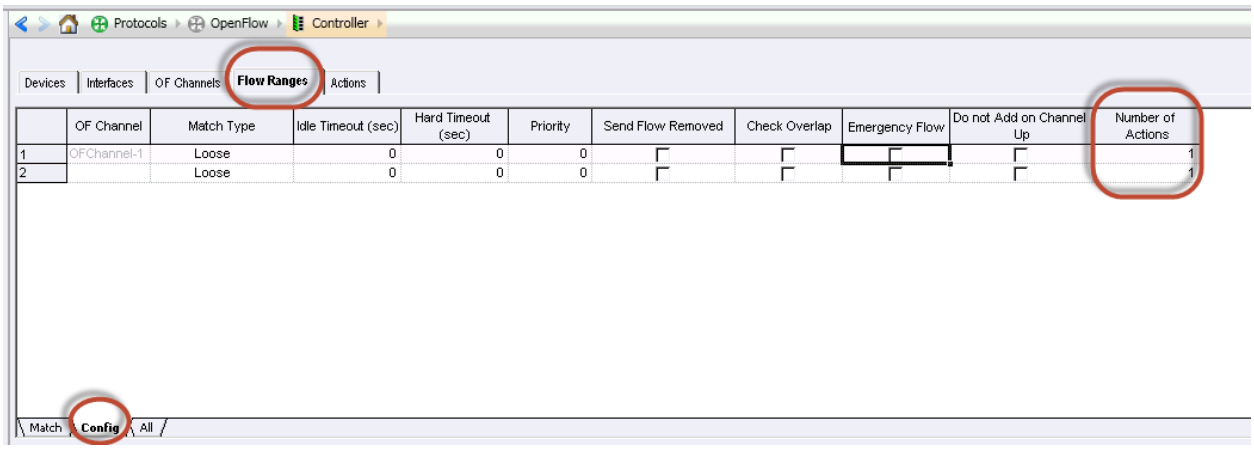


Figure 26: FlowRanges tab, OpenFlow window

9. Use **Configure Range** button (…) to increment MAC and VLAN.



10. Now create **Number of Action** from **Config** tab.



11. Go to **Actions** tab and select **Action Type** as *OutPut* and **Output Port Type** as *Custom/Manual*. Enter the **Output Port** value of the switch where the traffic will be forwarded to.

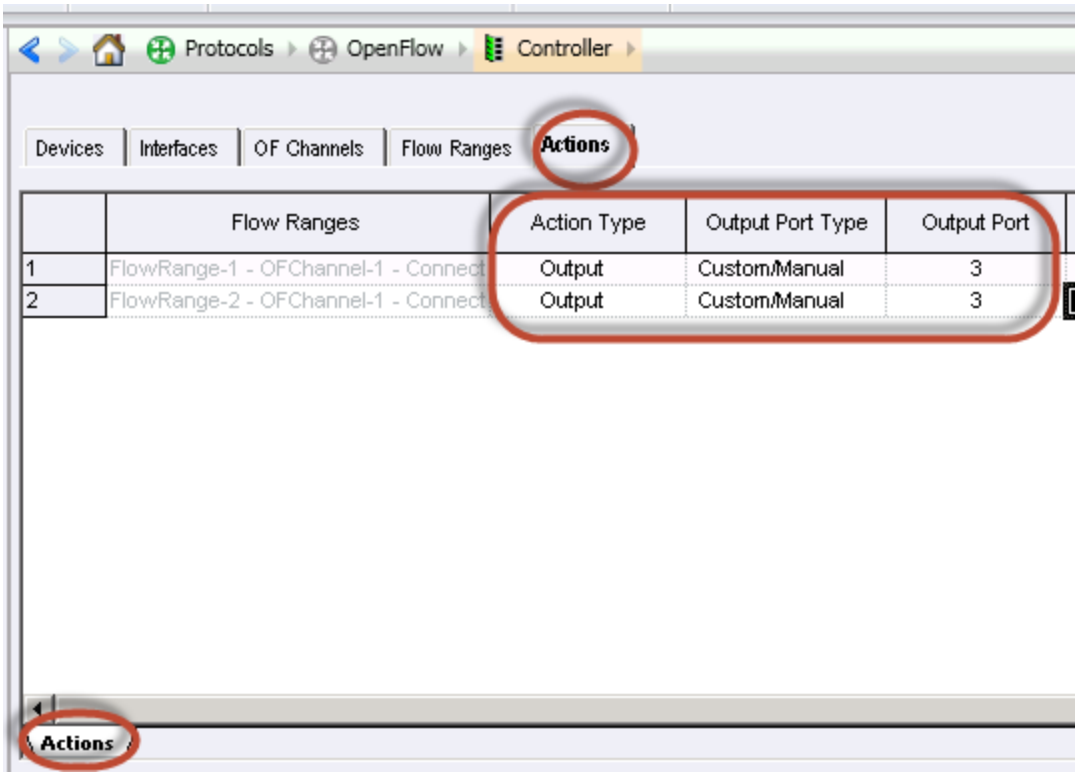


Figure 27: Actions tab

12. Use the **OpenFlow** control on the ribbon to start OpenFlow protocol and make sure OF Channel comes up. The value of the **OF Channel Configured UP** field indicates that the OF Channel is up.

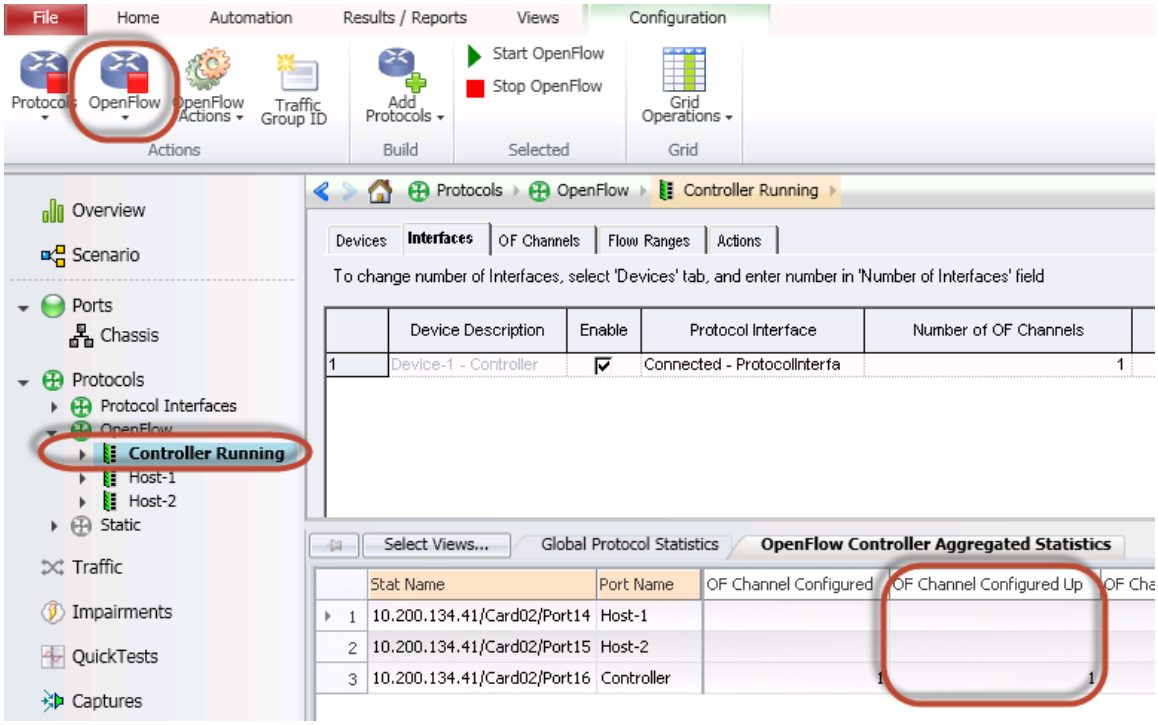


Figure 28: Controller Running, showing OF Channel Configured Up

To verify the switch capabilities, supported action or any error condition, go to **Learned Information** window. Several tabs are available as shown in the following figure. Click **Refresh** button on the ribbon to update the information.

Go to **OF Channel learned Info** tab, It has two panes. Left pane displays OF Channel information including **TCP Port**, **Data Path ID**, **Reply State** and any error message received from the Switch. When you select a row (OF Channel), the right pane displays all OpenFlow enabled ports information on that switch.

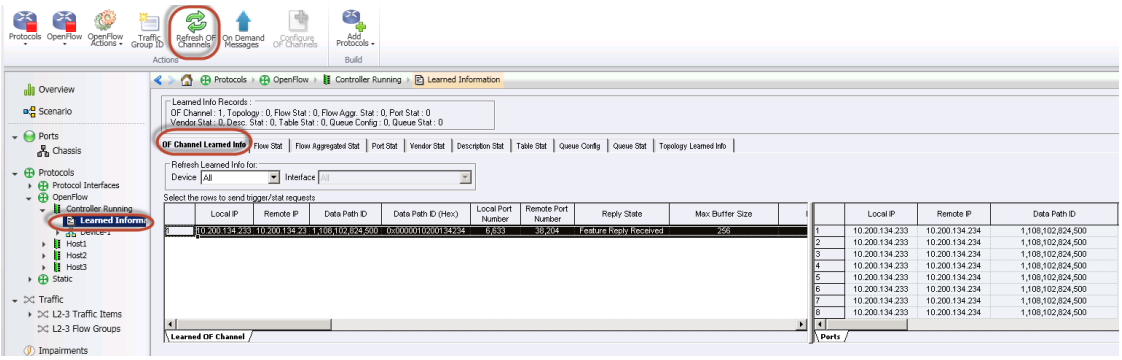


Figure 29: Learned Information window

13. From **OF Channel Learned Info** tab use **On Demand Messages** button to request switch to send flow table information.

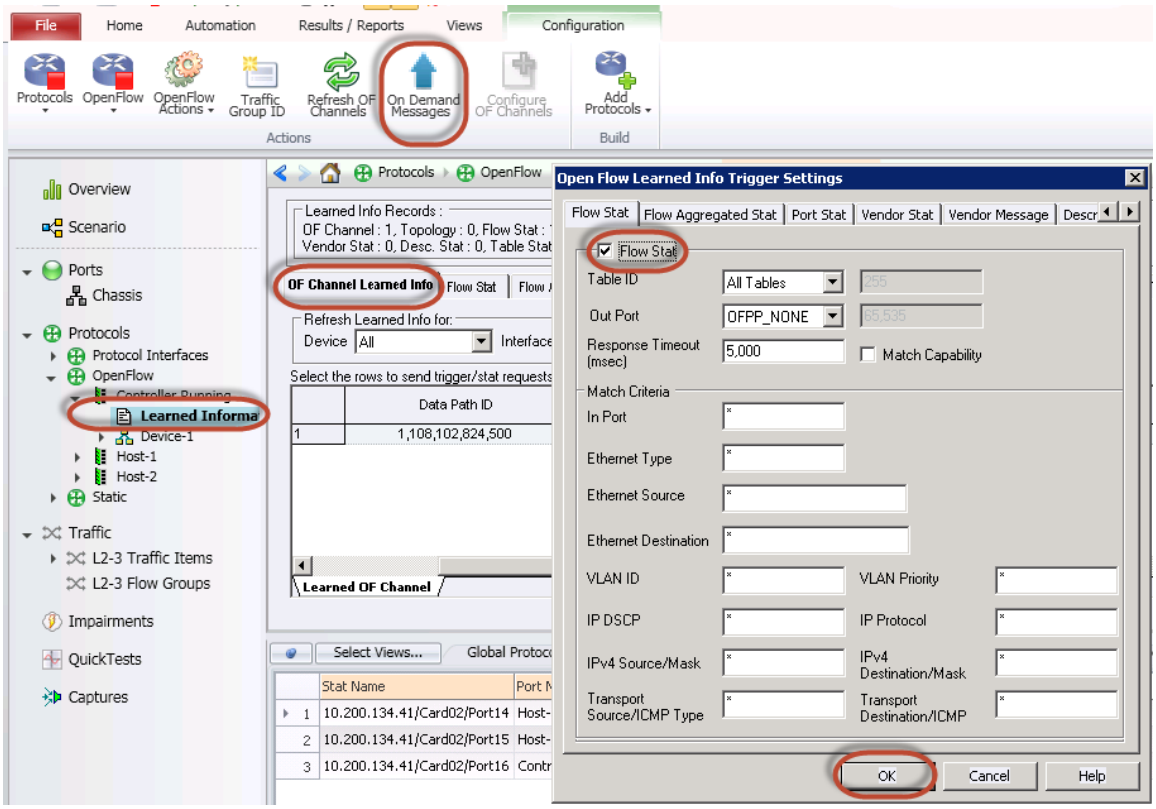


Figure 30: OpenFlow Learned Info Trigger Settings window

To verify flow table information, go to **Flow Stat** tab. On this tab, make sure that switch has correct flow entries to match the fields defined earlier in the flow range**,** input port and wild card entry for non-matching field.

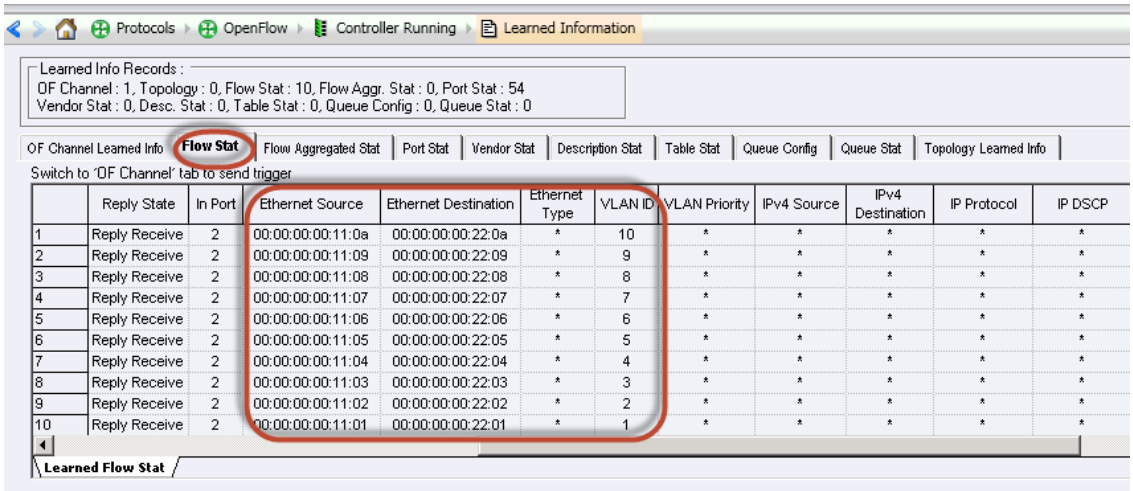


Figure 31: Flow Stat tab, Learned Information window

14. Create Traffic endpoints on host ports using **Generate Traffic Endpoint** wizard. This option will be available from **Flow Ranges** tab.

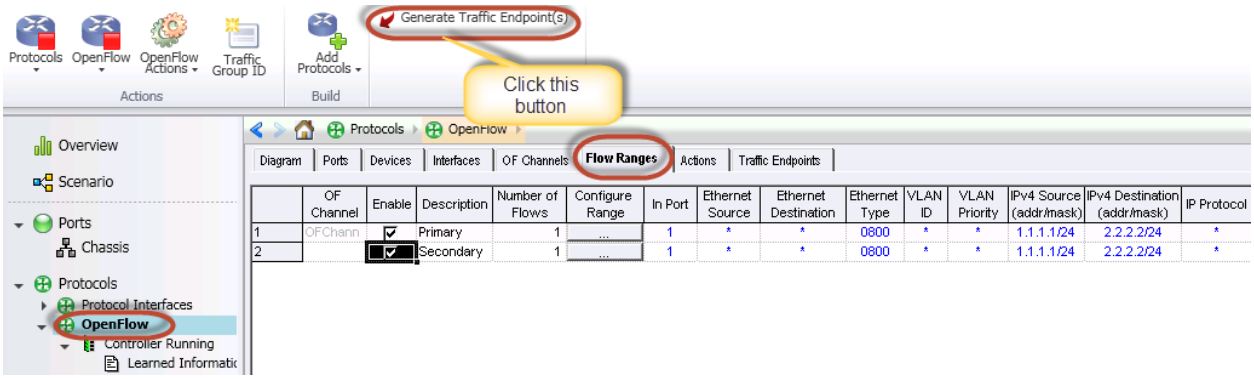


Figure 32: Flow Ranges tab

15. The following steps will help you use the **OpenFlow Traffic Converter Wizard** to create the corresponding traffic end points for the Flow Range values on Ixia ports.
    a. Select **host ports** where Traffic Endpoints will be created and click next.

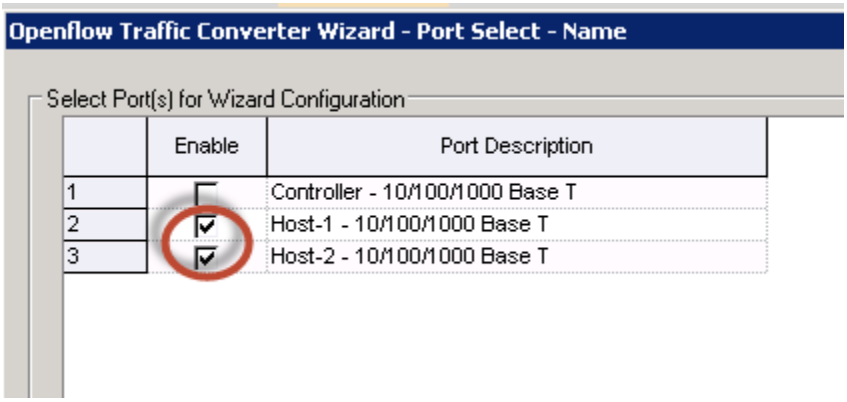
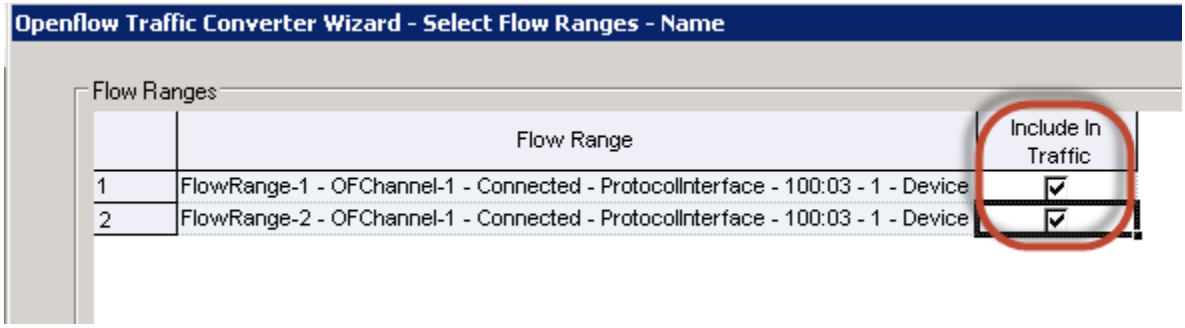
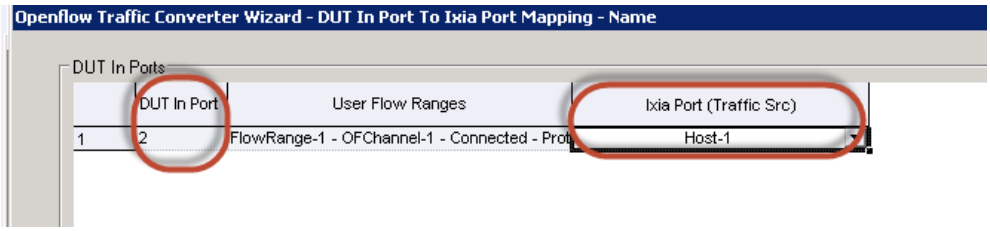


Figure 33: OpenFlow Traffic Converter Wizard

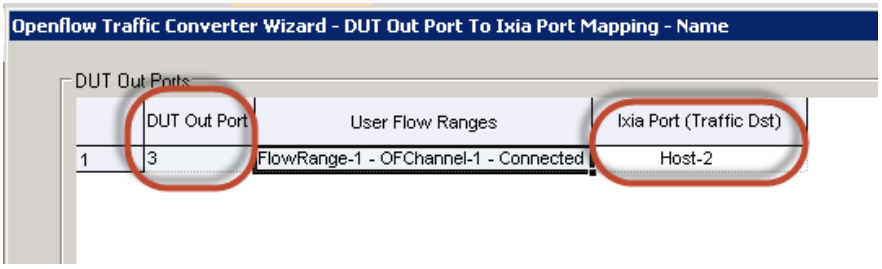
    b. Enable **Flow Ranges** to be included for traffic endpoints



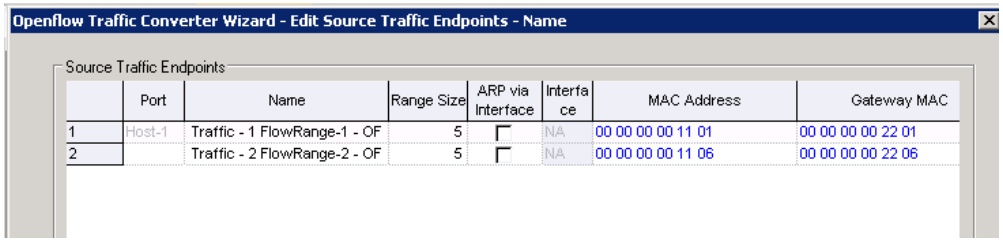
c. Map the Traffic source port, Host-1 in following figure with **DUT In port**. This will enable IxNetwork to map the traffic ports to switch ports.



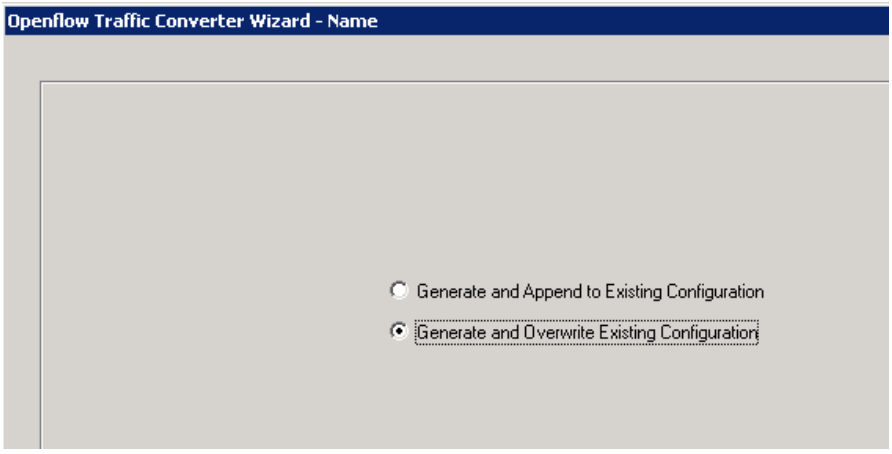
d. Map Traffic destination port, Host-2 in following figure with **DUT Out port**. This will enable IxNetwork to map the traffic ports to switch ports.



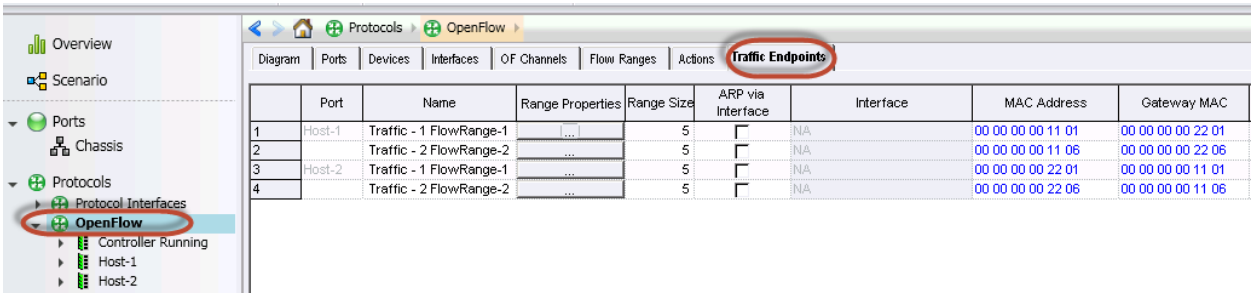
e. Leave everything default on next two windows.



f. Select **Generate and Overwrite Existing Configuration** option to remove previously generated traffic endpoint. Click Finish to complete the wizard configuration.



g. Go to each host port and make sure the wizard has generated correct traffic endpoint.



16. Go to **Traffic Wizard** to create **traffic flow** between **Host-1 and Host-2**

a. Select **Traffic** from the tree and click on **Add L2-3 Traffic** button in the ribbon. It opens the **Advance Traffic Wizard**.

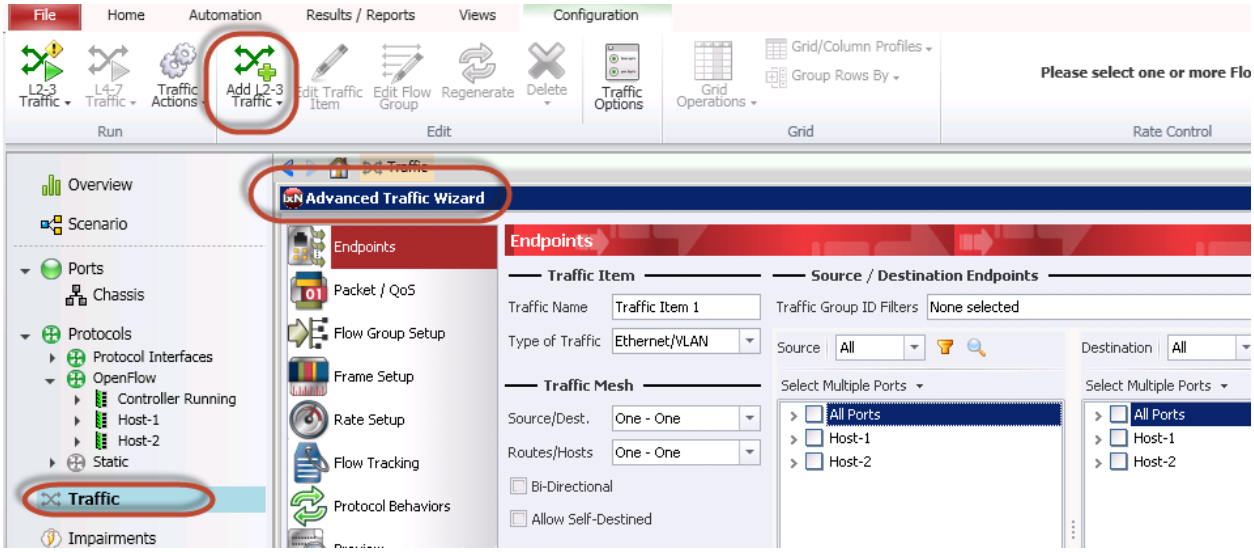
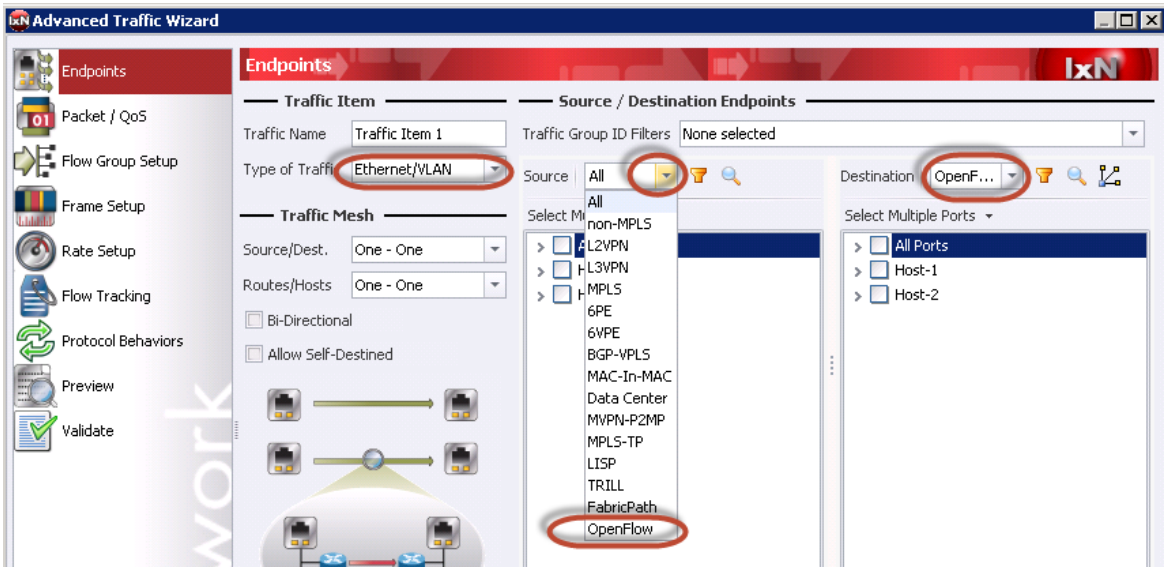


Figure 34: Advanced Traffic Wizard

b. Select **Type of Traffic** as **Ethernet/VLAN** and use **OpenFlow** encapsulation filter for **Source** and **Destination** endpoints.



c. Select source and destination endpoint and click Next.

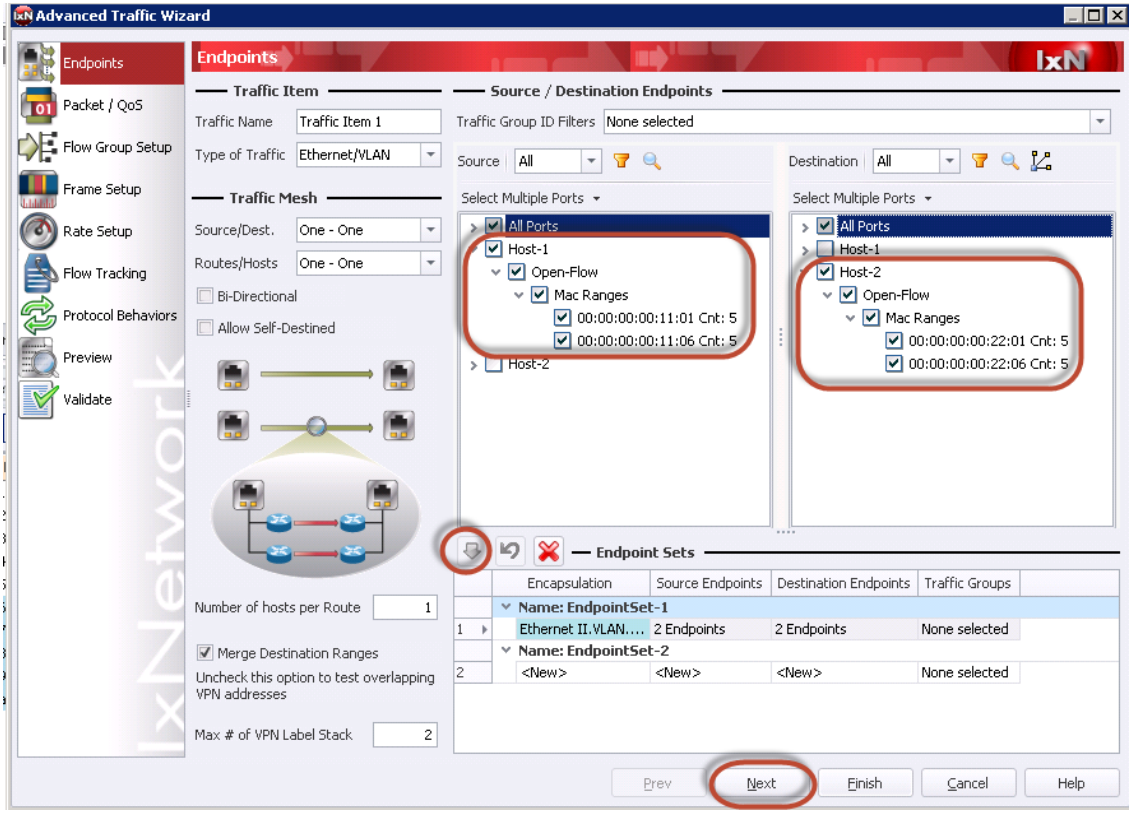
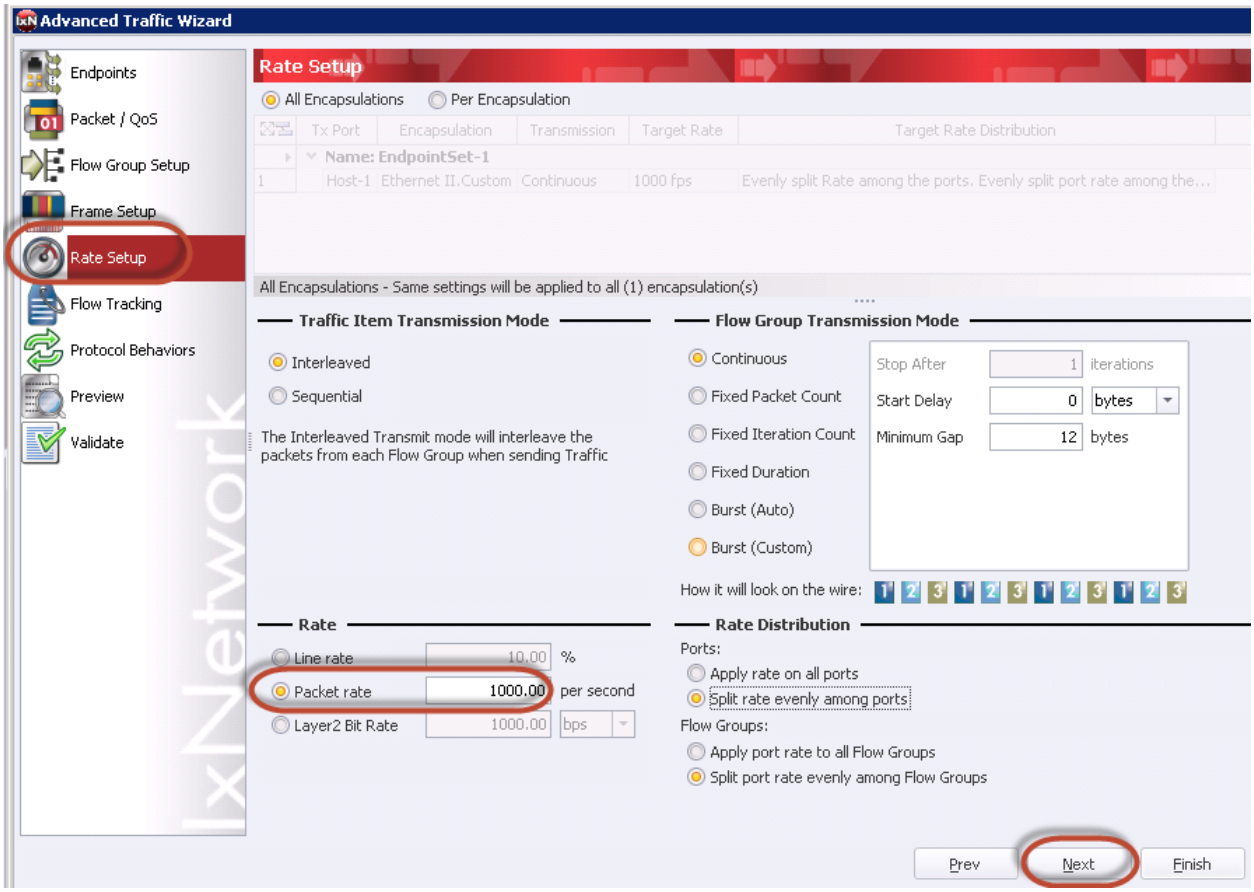


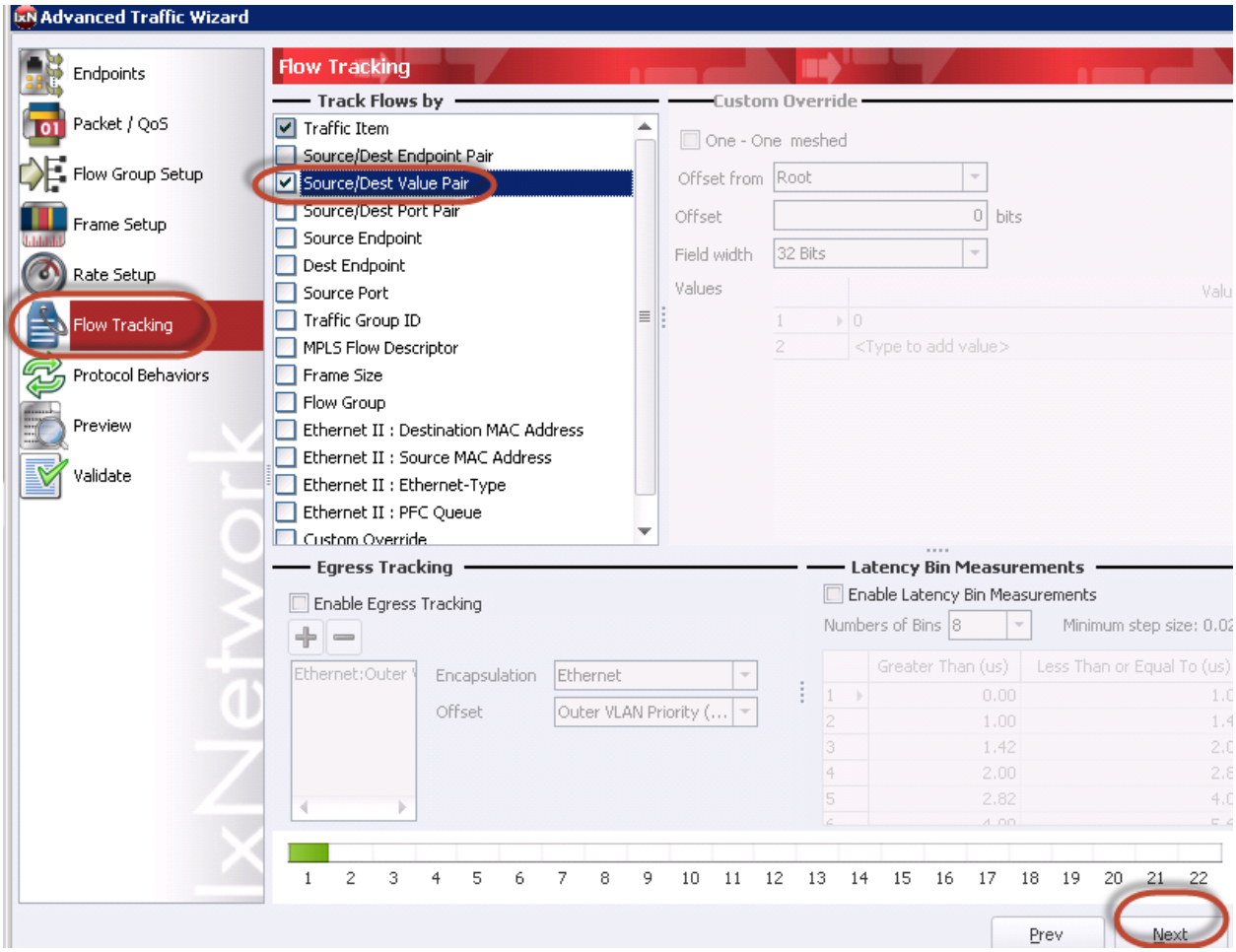
Figure 35: Advanced Traffic Wizard, Source and Destination Endpoints

d. Leave default values for **Packet/QoS**, **Flow Group setup** and **Frame Setup** page.

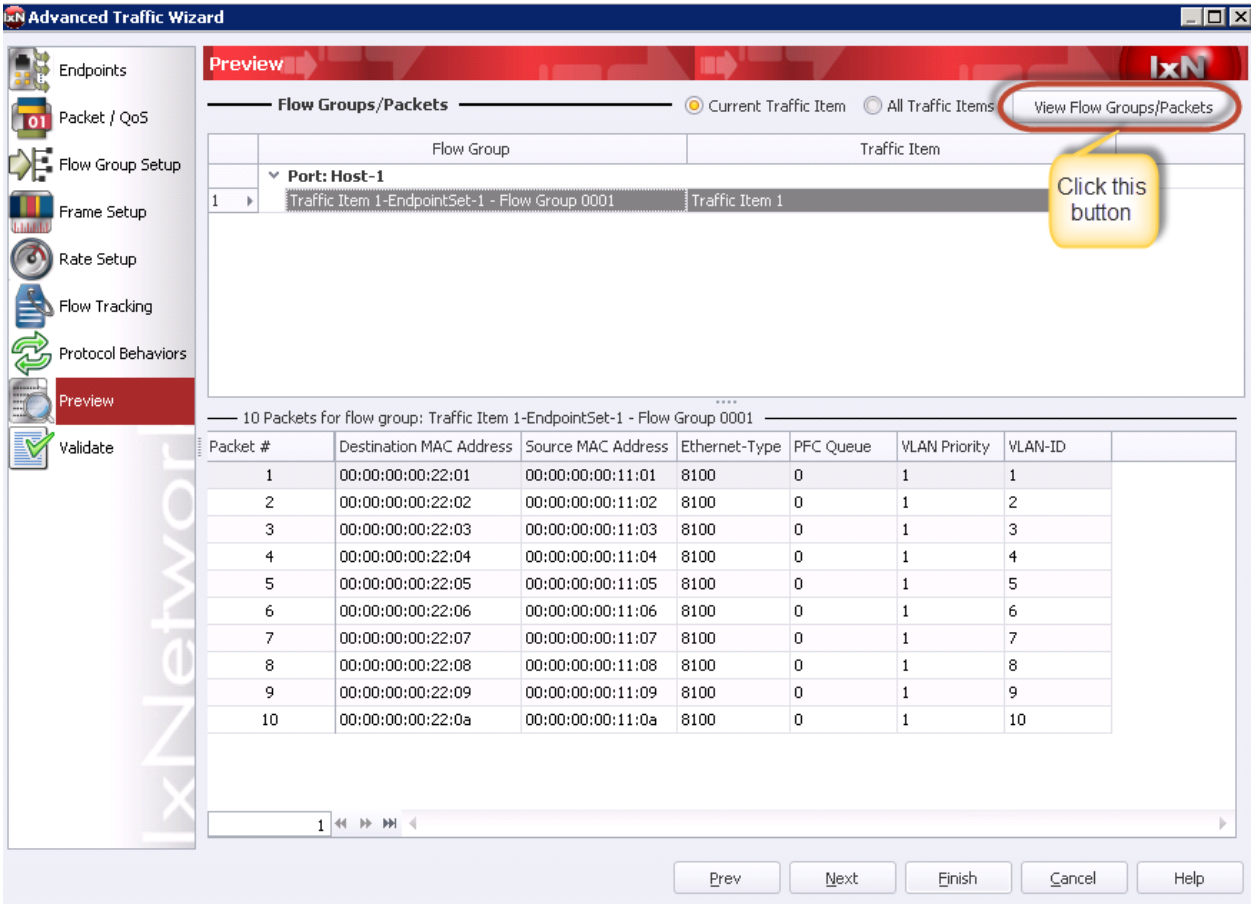
e. Set the desired traffic load on **Rate Setup** page and click **Next**.



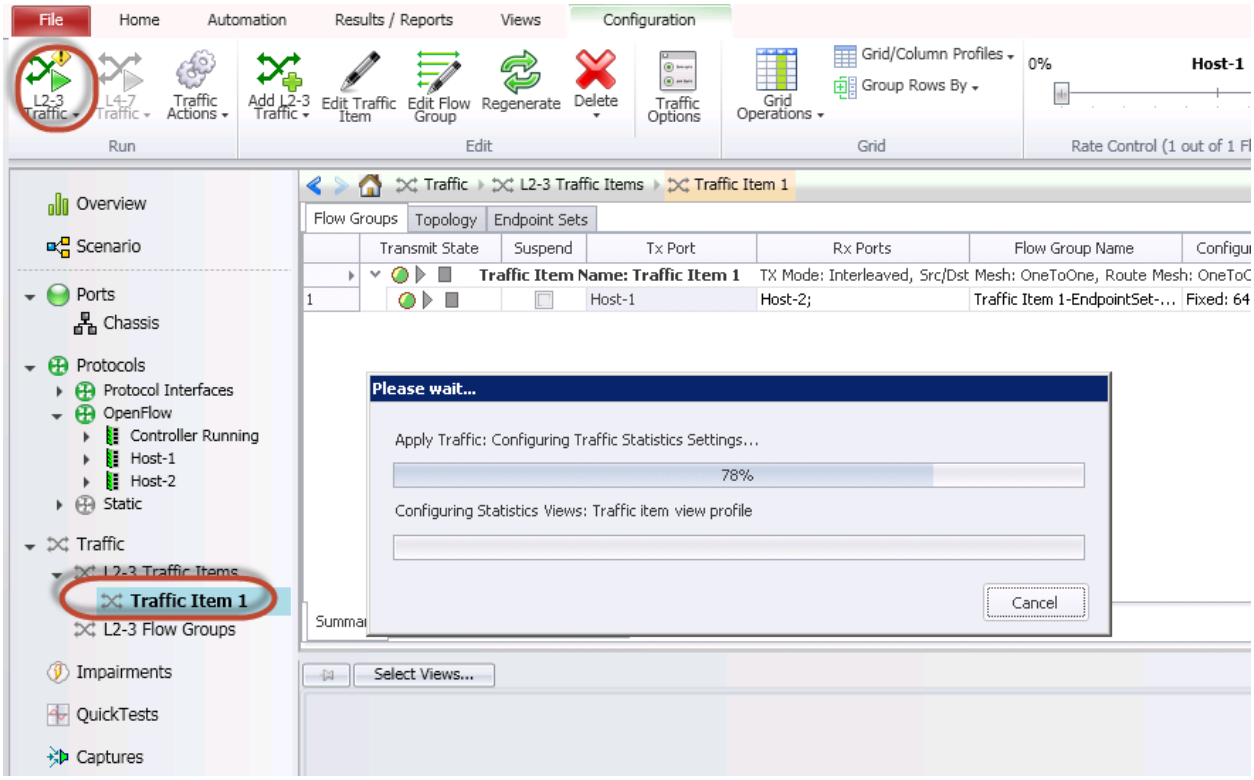
f. On **Flow Tracking** page, enable **Source/Dest Value Pair** tracking option. Click **Next**.



g. Skip '**Protocol Behaviors** window and go to **Preview** window to view how the traffic flow will look like and click **Finish** button to end traffic wizard.



17. Click **L2-3 Traffic** button to push the traffic on port and start traffic.



Results Analysis

- On **Traffic Item Statistics** verify that traffic is flowing through the switch without packet loss.

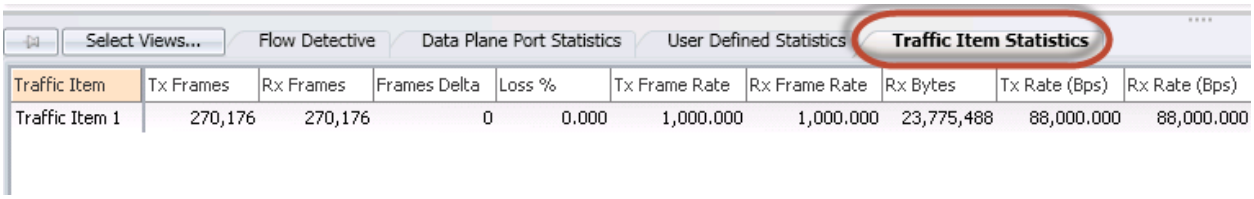
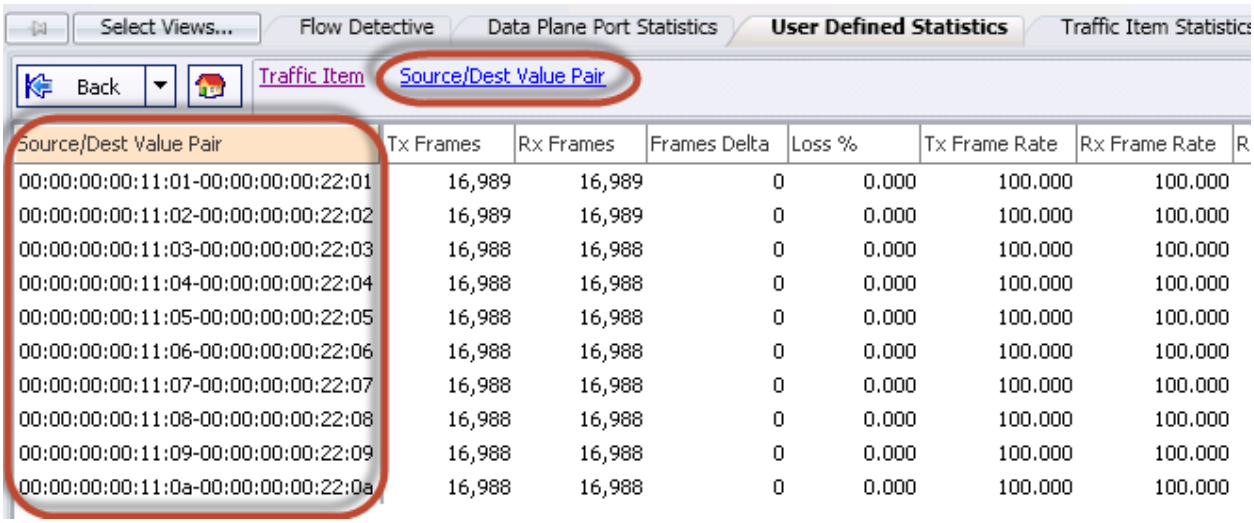


Figure 36: Traffic Item Statistics view

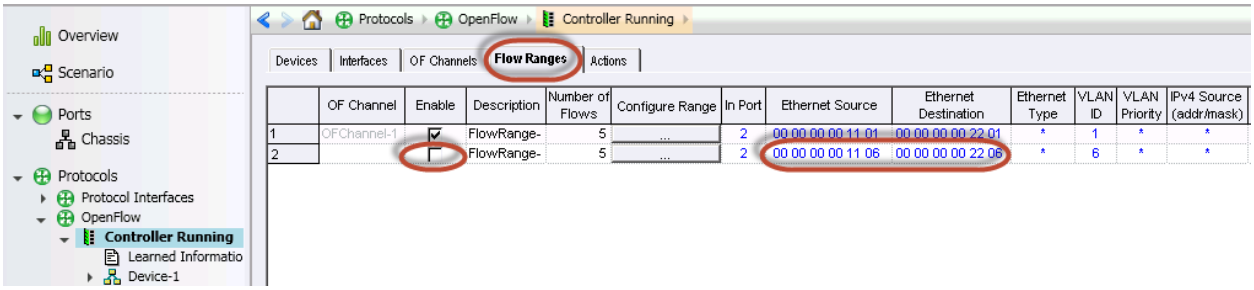
- Select the **Traffic Item** and Right click on it. Select drill down per **Source/Destination Value Pair** tracking option. This drill down view will display traffic statistics per individual source and destination MAC address.



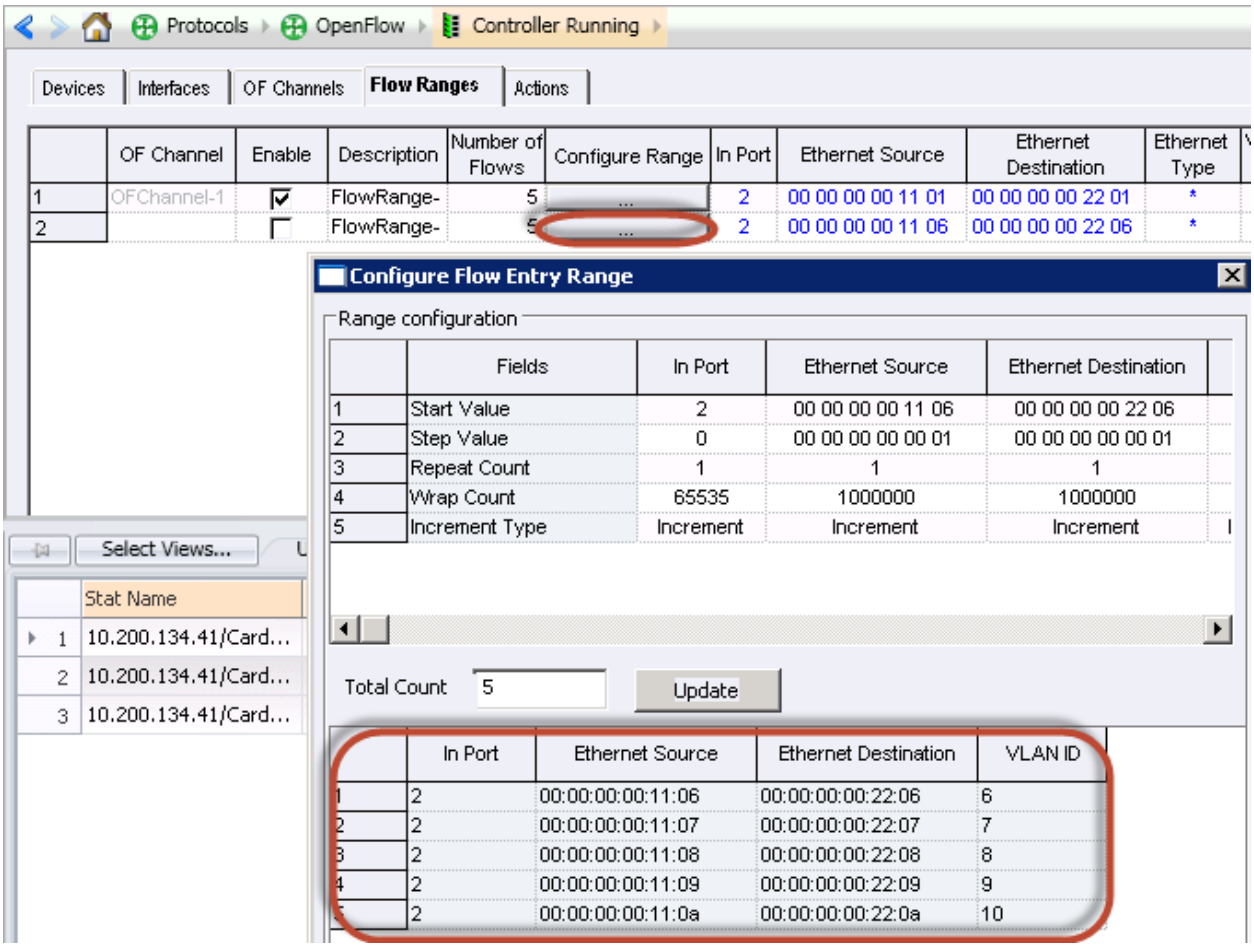
Source/Dest Value Pair	Tx Frames	Rx Frames	Frames Delta	Loss %	Tx Frame Rate	Rx Frame Rate	R
00:00:00:00:11:01-00:00:00:00:00:22:01	16,989	16,989	0	0.000	100.000	100.000	
00:00:00:00:11:02-00:00:00:00:00:22:02	16,989	16,989	0	0.000	100.000	100.000	
00:00:00:00:11:03-00:00:00:00:00:22:03	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:04-00:00:00:00:00:22:04	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:05-00:00:00:00:00:22:05	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:06-00:00:00:00:00:22:06	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:07-00:00:00:00:00:22:07	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:08-00:00:00:00:00:22:08	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:09-00:00:00:00:00:22:09	16,988	16,988	0	0.000	100.000	100.000	
00:00:00:00:11:0a-00:00:00:00:00:22:0a	16,988	16,988	0	0.000	100.000	100.000	

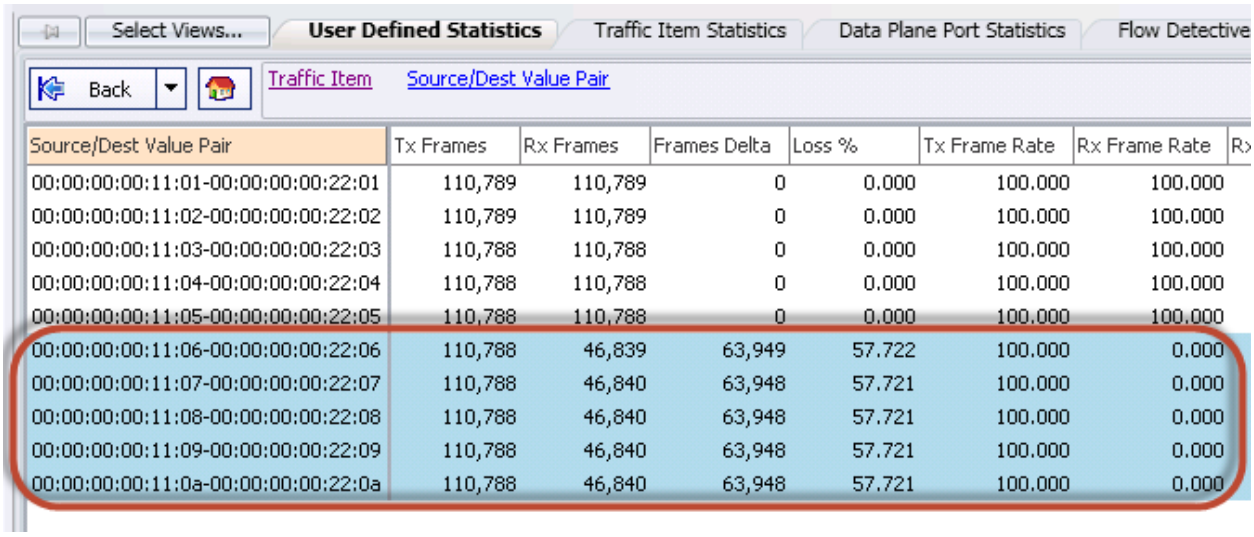
- Make policy change on controller and push it to the switch and verify that switch changes packet forwarding decision according to the rule set by the controller.

  Leave traffic and OpenFlow protocol in *Running* state. Go to **OpenFlow Controller Flow Range** and clear the second flow range.

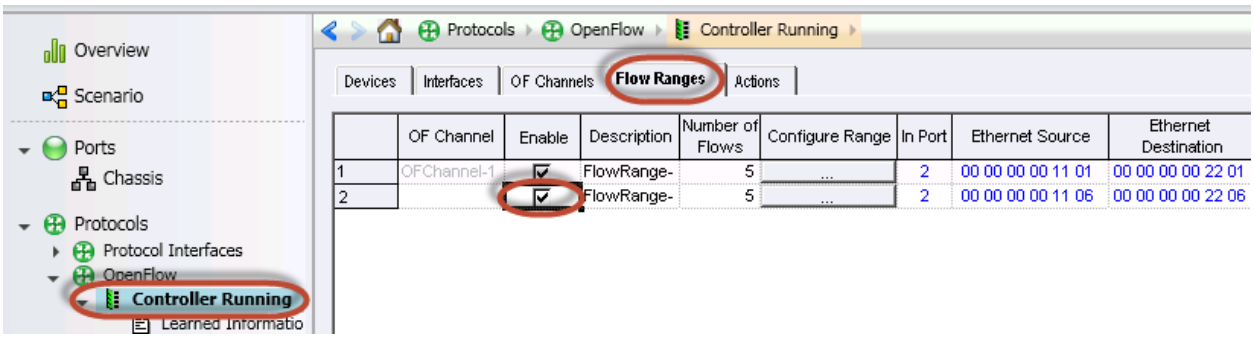


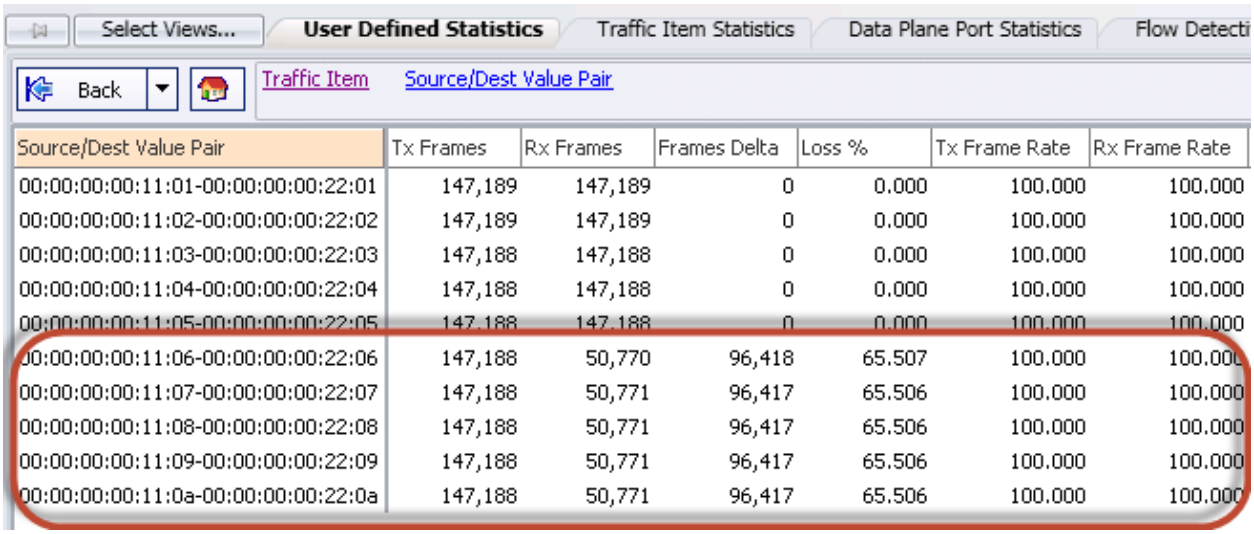
Ensure following MAC address and VLAN stops receiving traffic. Go to traffic statistics to verify this functionality.





- Select the flow range again and see if switch starts forwarding traffic for that MAC/VLAN again.





Conclusions

This test case can be used to verify:

- Switch installs correct flow entries in its flow table as pushed by the controller.
- It makes packet forwarding decision as per the rule set by the controller.
- It complies to one or more action set by the controller.

Test Variables

The following variables can be used to verify the behavior of an OpenFlow Switch.

1. **L3 Forwarding Test**
   Set the matching criteria on L3 header field such as Source IP, Destination IP and DSCP value

2. **Use multiple Actions**

Apply multiple actions for each Flow range. Use one of the following actions along with Output and verify that switch correctly performs multiple actions on the packets.

Set VLAN ID	Add or Change VLAN ID for matching flow
Strip VLAN Header	Strip the VLAN header from the matching flow
Set Ethernet Src/Dst.	Change source/destination MAC for matching flow
Set IP DSCP	Change DSCP value for matching flow
Set IPv4 Src/Dst.	Change IP address for matching flow
Set Transport Src/Dst.	Change TCP/UDP port

Test Case: Switch Flow Failover Performance Test

Overview

Networking infrastructure has become key component for any business. Today's networks carries Voice, Data and Video traffic over the same network infrastructure. Even the few seconds outage can cause huge impact on the business. Therefore networks are designed with redundant links to minimize the downtime and increase the reliability. The key challenge for any networking device is to be able to detect the failure and forward traffic to redundant path without affecting application traffic performance.

Objective

The objective of this test case is to verify Switch performance to handle convergence in the network and ability to converge the traffic to secondary path without affecting traffic forwarding performance. For this test, Ixia's unique TrueView convergence test methodology will be used to measure the data-plane to data-plane convergence time of the switch.

TrueView convergence test provides the following measurements:

- Timestamp of every packet
- Timestamp the first packet in and last packet out on a port per flow
- Ability to capture protocol event timestamp
- Ability to capture link event timestamp
- Ability to monitor receive rate and timestamp when "Below" thresholds are crossed
- Ability to monitor receive rate and timestamp when "Above" thresholds are crossed
- Ability to timestamp link event (up/down)

The diagram below shows, when traffic drops below threshold value on primary path (Blue line), it will latch tDP-below1 timestamp and when traffic reaches above threshold value on a secondary path (Red line), it will latch tDP-above2 timestamp. This timestamp will give data-plane convergence time.

DP-DP Convergence time = tDP Above timestamp – tDP Below timestamp

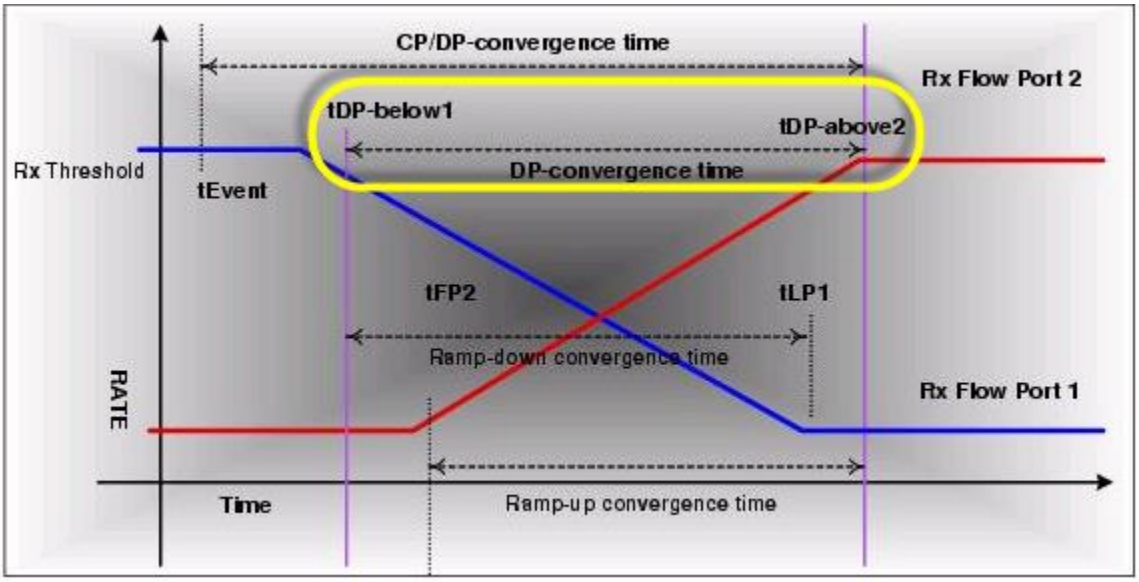


Figure 37: CP/DP Convergence Time, calculation

Setup

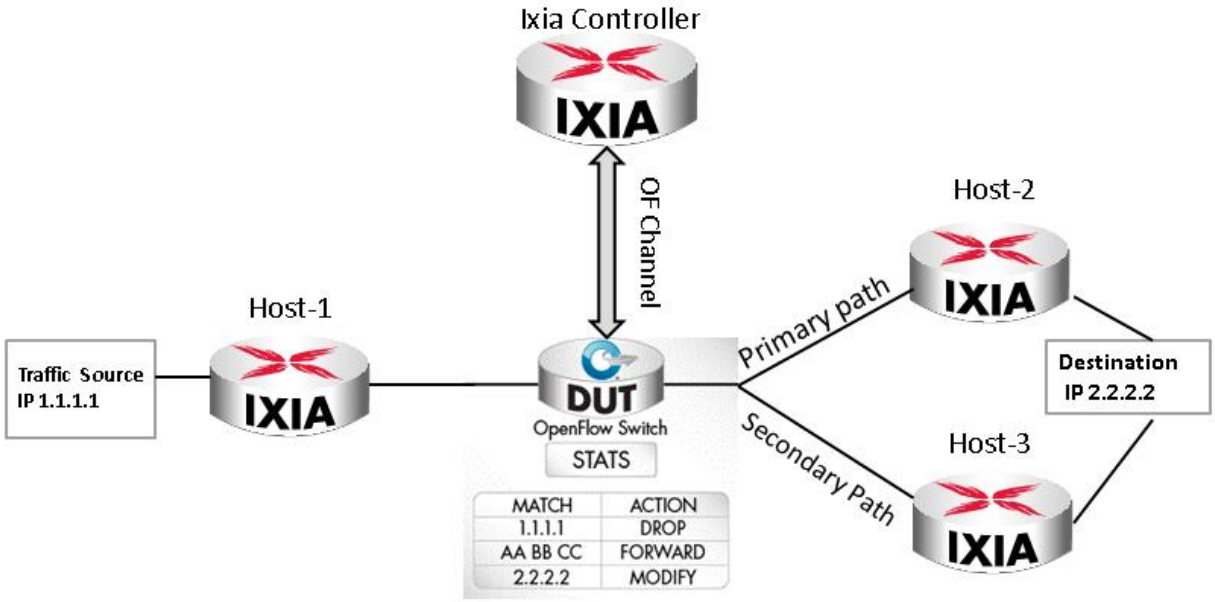


Figure 38: Switch Flow Failover Performance Test, test setup

Step-by-Step Instructions

1. Reserve 4 Ixia port (1 for controller and 3 for data-plane traffic)

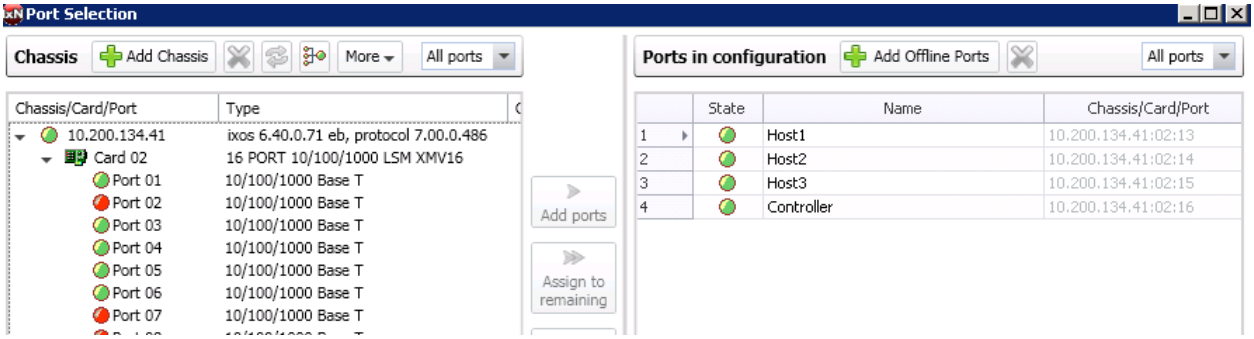


Figure 39: Port Selection window

2. Enable OpenFlow on all 4 ports by selecting the OpenFlow check box on the RoutingSwitching tab on the Protocols window.

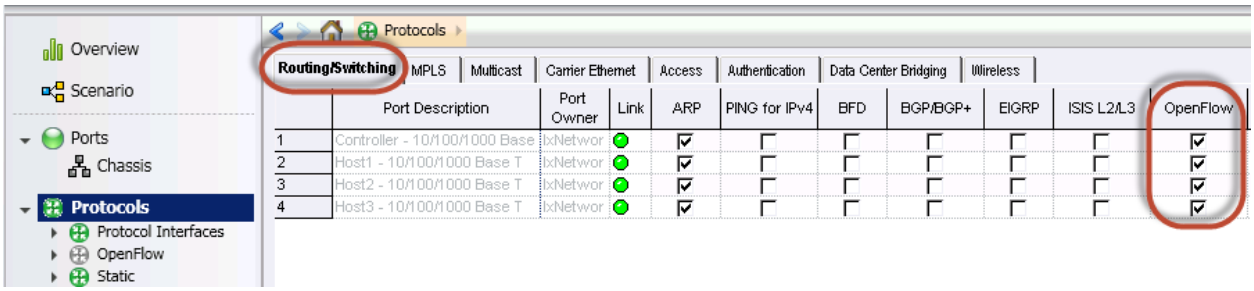


Figure 40: RoutingSwitching tab, Protocols window

3. Go to **Connected Interface** tab to configure Emulated Controller IP Address and **Gateway Address**.

   Use OpenFlow switch's IP address if you have only one switch. Ensure that ARP is resolved for for OF Channel. Do not configure anything on host ports.

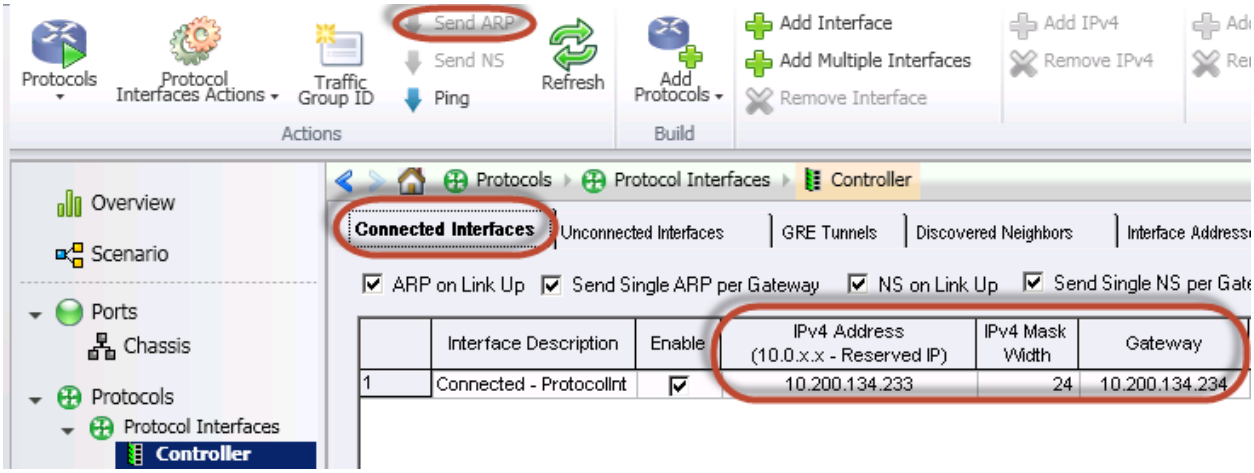


Figure 41: Connected Interface tab

4. Go to Ports tab and select the **Port Role**.

   You can select any of the following:

   a. **Control**: for Controller port
   b. **Traffic**: for host ports

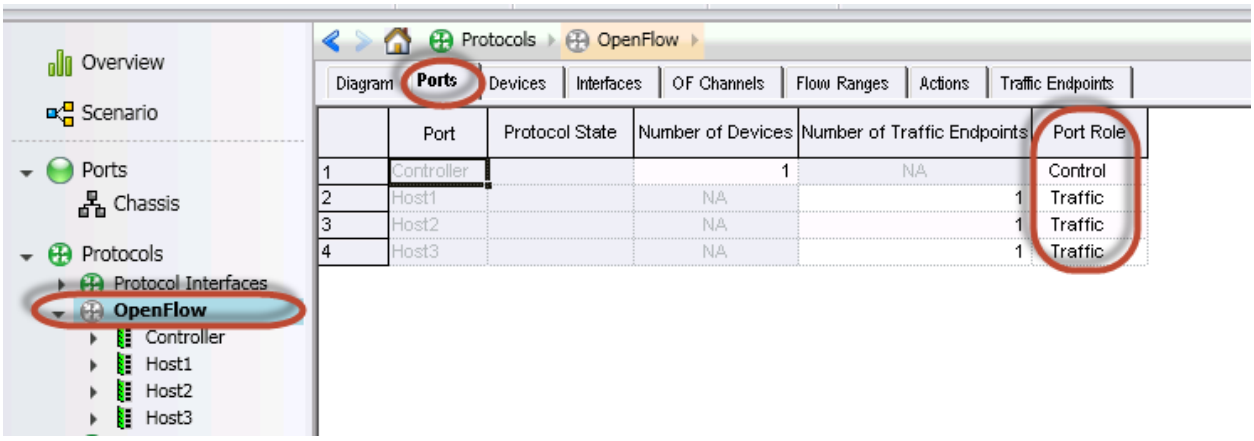


Figure 42: Ports Tab

5. Go to **Devices** tab and configure **Number Of Interfaces** as *1*. The number of interfaces equals the number of emulated NICs of a controller.

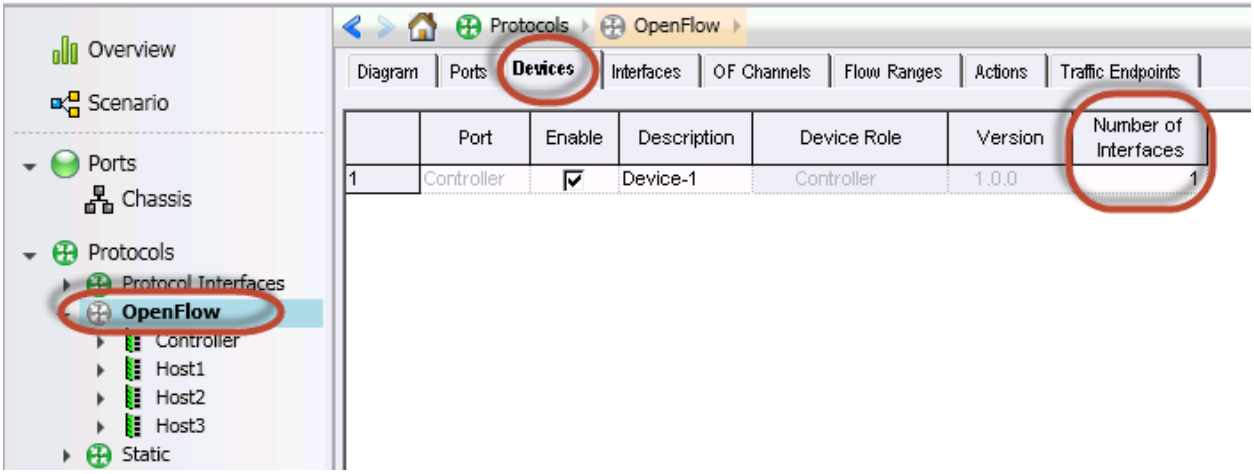
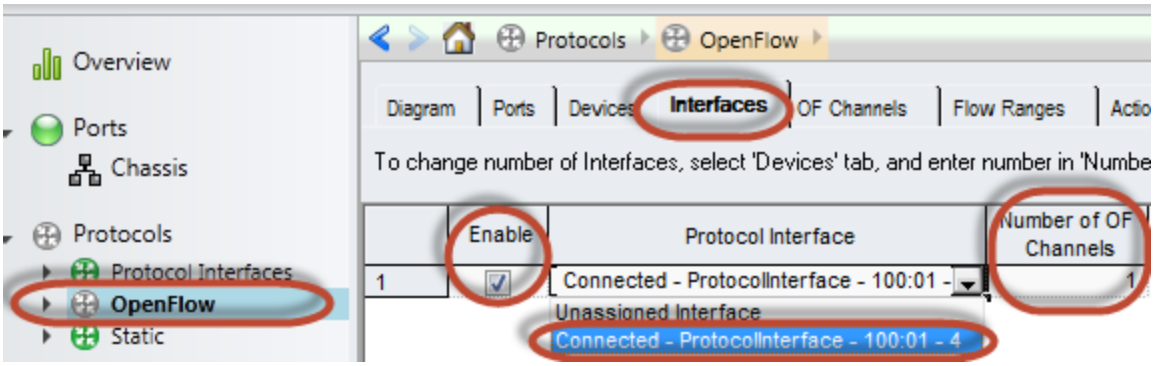
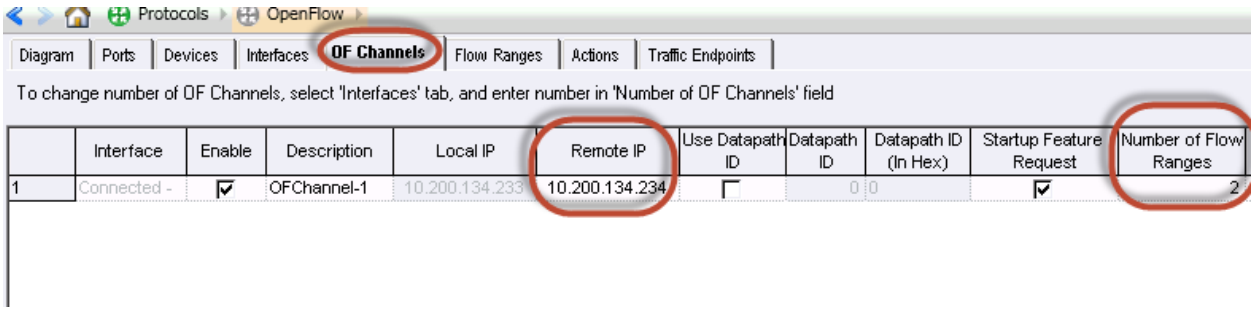


Figure 43: Devices tab

6. Go to **Interfacees** tab and assign the interface previously created under protocol interface. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1*.



7. Go to **OF Channels** tab and enter DUT IP address in **Remote IP** field. Enter Number of Flow Ranges as 2.



8. Under **Flow Ranges** tab, configure **Ethernet Type**, Source and Destination IP address. Use wild card (*) for remaining all fields. Use same IP address for both ranges. Configure correct **In Port** field with the switch port number that is connected to Ixia host port generating traffic.

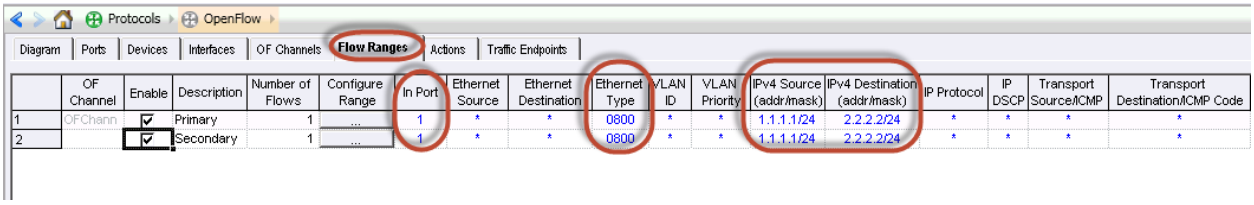


Figure 44: Flow Ranges tab

9. Create **Number of Action** from config tab. Change **Match Type** to *Strict* and **Priority** value for both flows. The flow with higher priority value gets forwarded first.

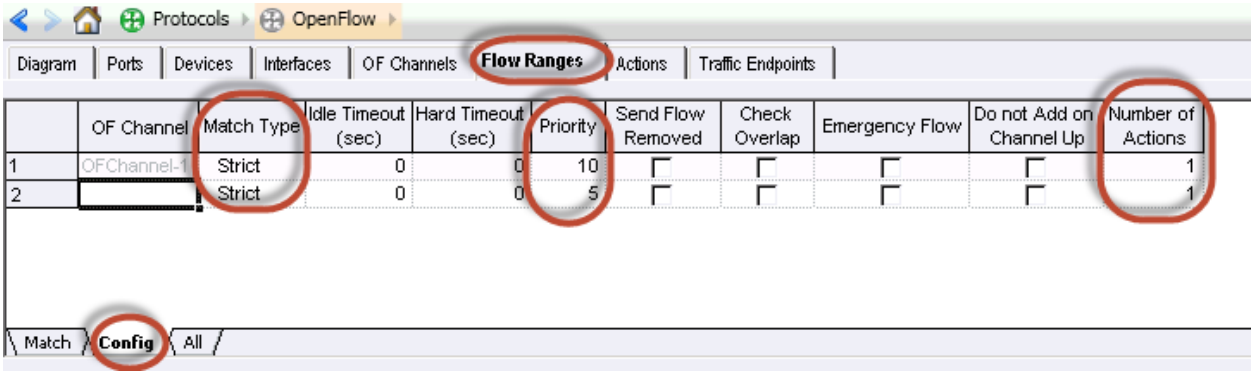


Figure 45: Flow Ranges tab (2),

10. Go to **Actions** tab and select **Action Type as** *OutPut* and **Output Port Type** as *Custom/Manual*. Enter the **'Output Port'** value of the switch where the traffic will be forwarded to.

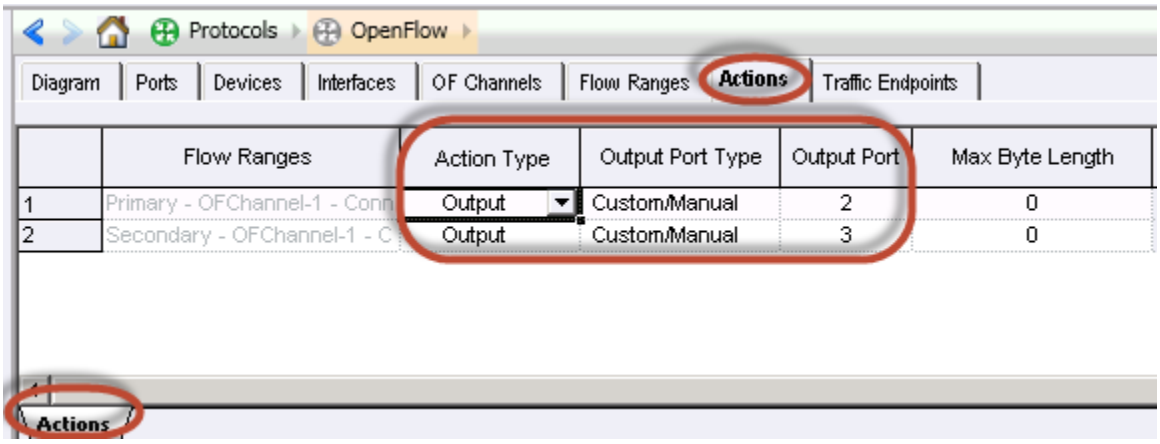


Figure 46: Actions tab

11. Start OpenFlow protocol using the OpenFlow control on the ribbon, and make sure OF Channel comes up.

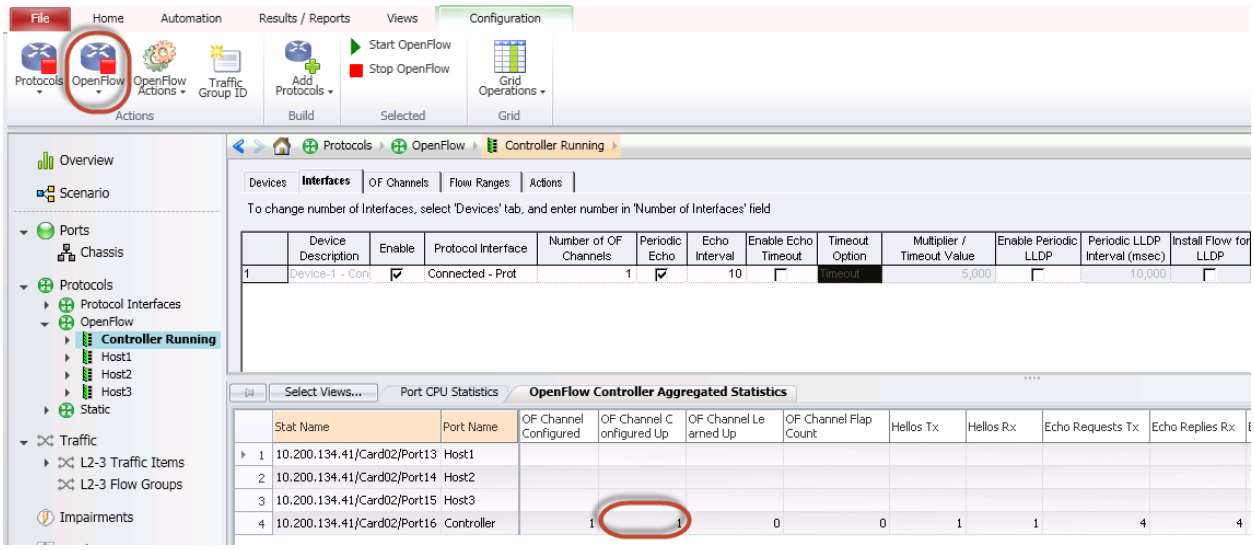


Figure 47: Controller Running window

To verify the switch capabilities, supported action or any error condition, go to **Learned Information** window. Several tabs as shown in below figure are available on this window. Click **Refresh** button in the ribbon to update this information. Go to **OF Channel Learned Info** tab. It displays several panes. The left pane displays OF Channel information including TCP port, Data path ID, Reply state and any error message received from the Switch. When you select a row (OF Channel), the right pane displays all OpenFlow enabled ports information on that switch.

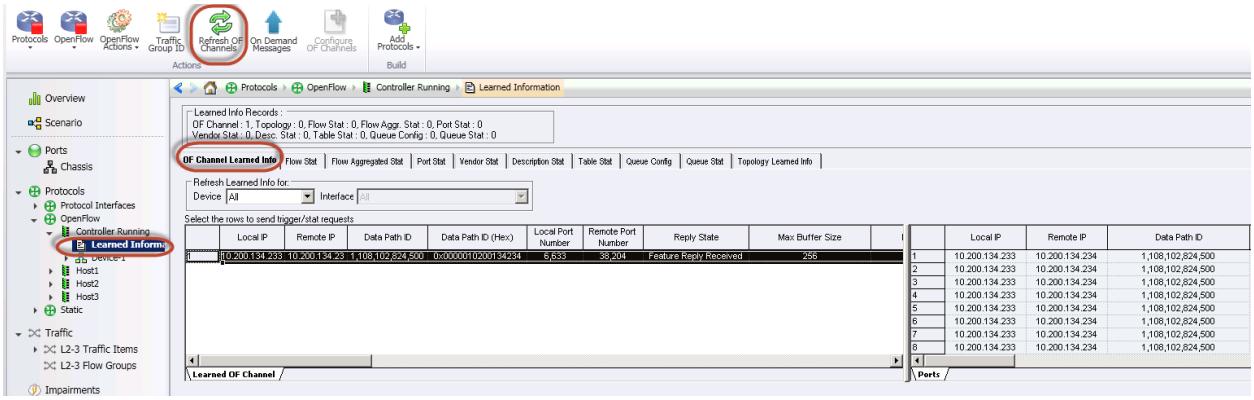


Figure 48: OF Channel Learned Info

From **OF Channel Learned Info** tab use **On Demand Messages** button to request switch to send flow table information.

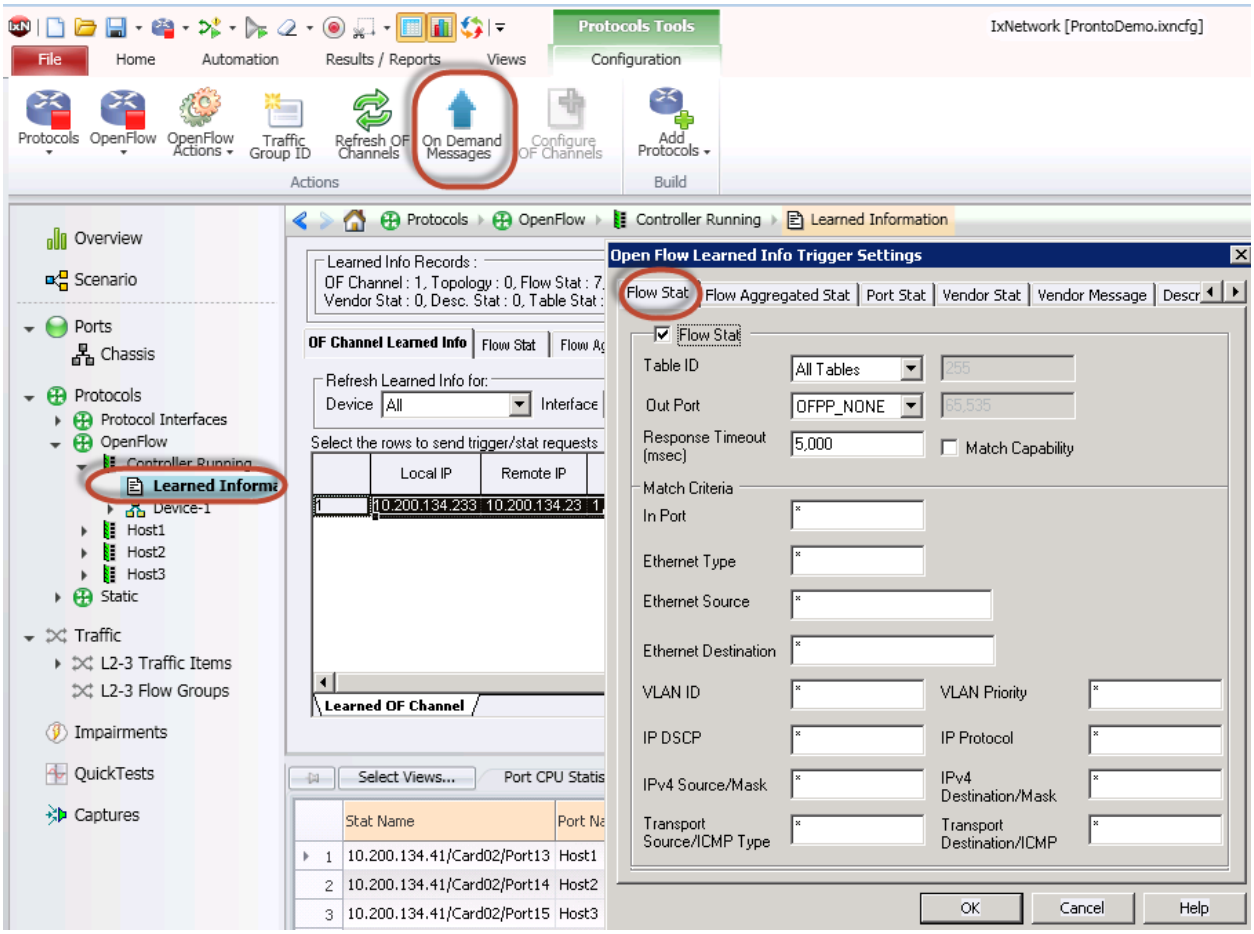
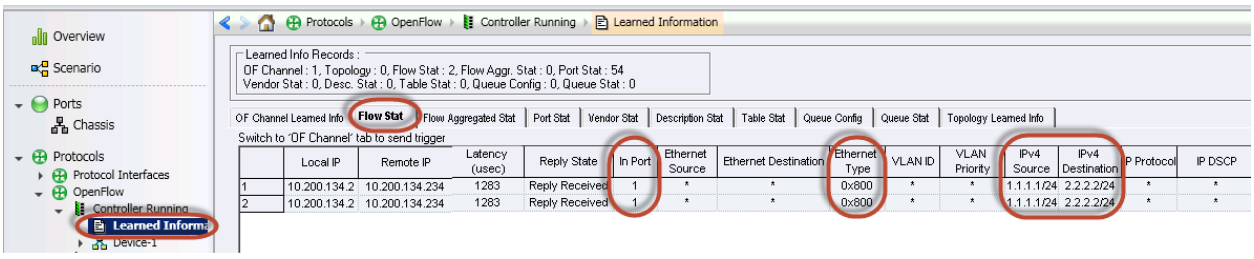


Figure 49: OpenFlow Learned Info Trigger Settings window

To verify flow table information, go to **Flow Stat** tab and ensure that switch has correct flow entries to match the fields defined earlier in the flow range. Enter port and wild card entry for non-matching field.



Create Traffic endpoints on host ports using the **Generate Traffic Endpoint** wizard. This option is available on the **Flow Ranges** tab.

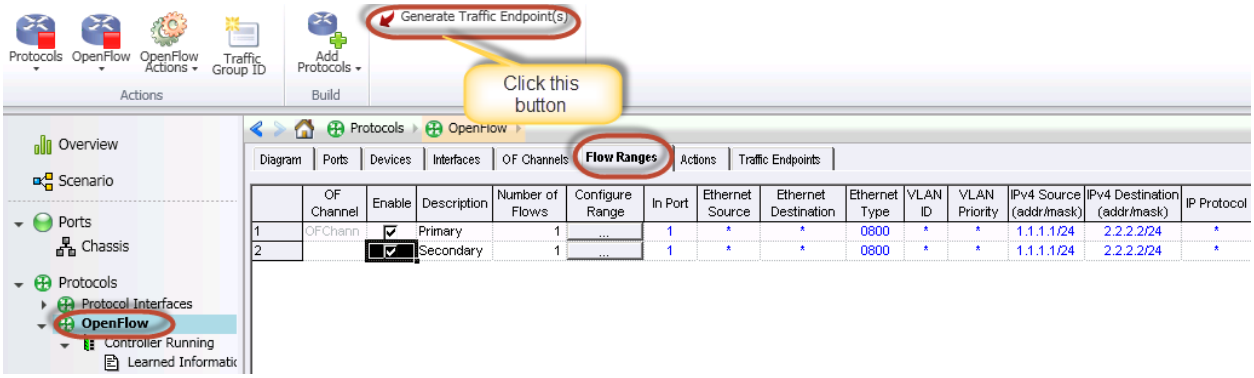


Figure 50: Flow Ranges tab

The following steps will help you use the **OpenFlow Traffic Converter Wizard**. This will create the corresponding traffic end points for the Flow Range values on Ixia ports.

a. Select host ports where Traffic Endpoints will be created and click Next.

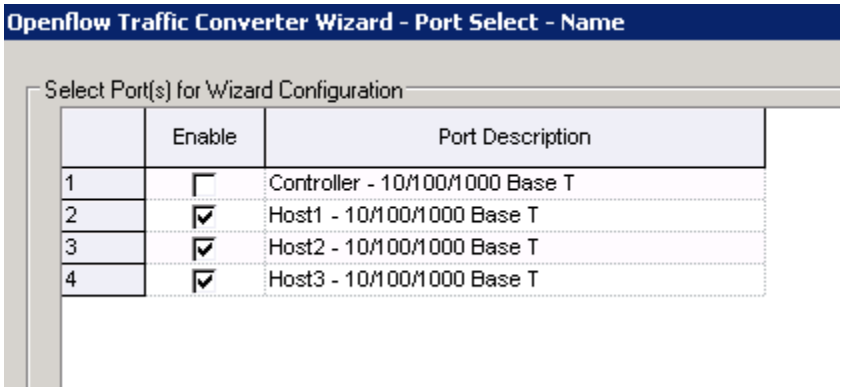


Figure 51: OpenFlow Traffic Converter Wizard – Port Select – Name window

b. Select both Primary/Secondary flow range to be included in traffic.

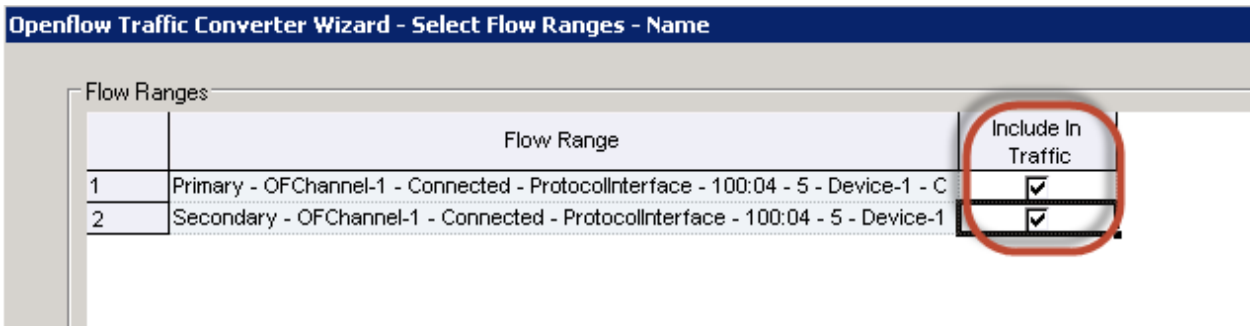


Figure 52: OpenFlow Traffic Converter Wizard – Select Flow Ranges - Name

c. Map the Traffic source port in the following figure *Host-1*, with DUT In port.

This will enable IxNetwork to map the traffic ports to switch ports

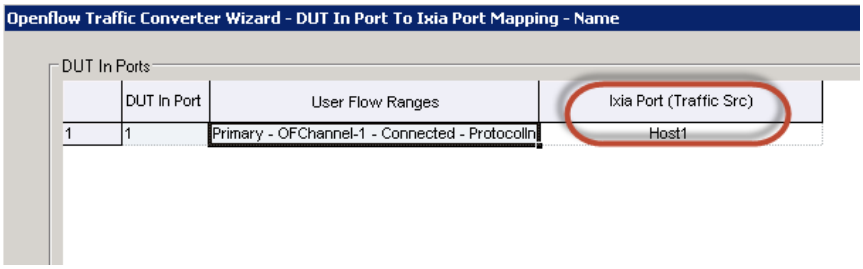


Figure 53: OpenFlow Traffic Concerter Wizard – DUT In Port To Ixia Port Mapping – Name window

d. Map traffic receiving ports with DUT's output port.
This will enable IxNetwork to map the traffic ports to switch ports

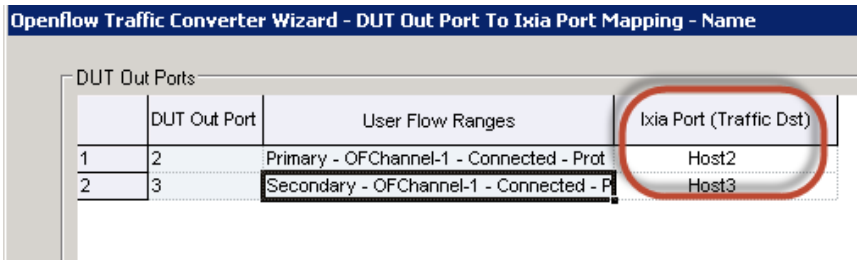


Figure 54: OpenFlow Traffic Converter Wizard – DUT Out Port To Ixia Port Mapping – Name window

e. Leave everything default on next two windows.
f. Select **Generate and Overwrite Existing Configuration** check box to remove previously generated traffic endpoint and click on finish.

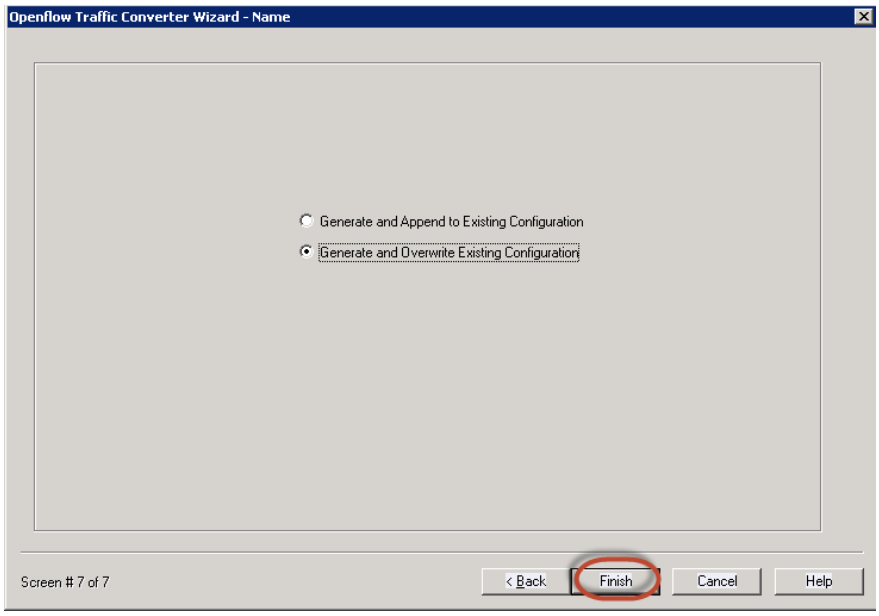
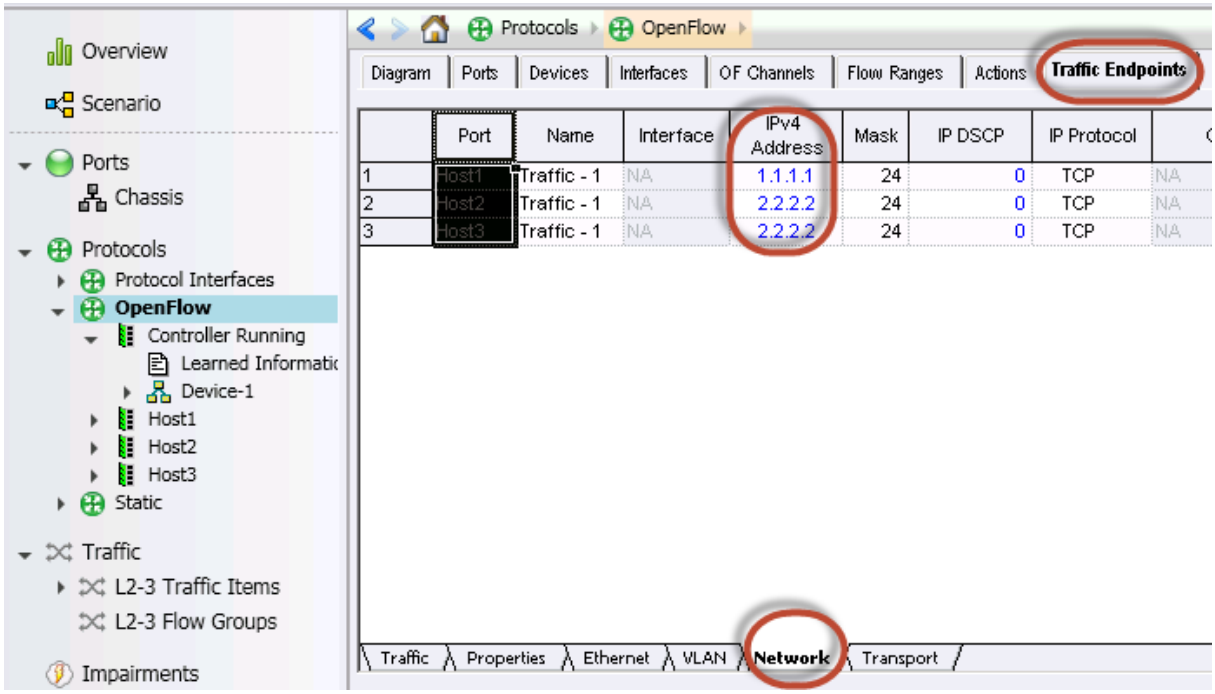


Figure 55: OpenFlow Traffic Converter Wizard – Name window

g. Go to each host port and ensure the wizard has generated correct traffic endpoint



12. Go to **Traffic Options** window and select **Data Plane Event Monitor** check box and set desired data-plane threshold limit.

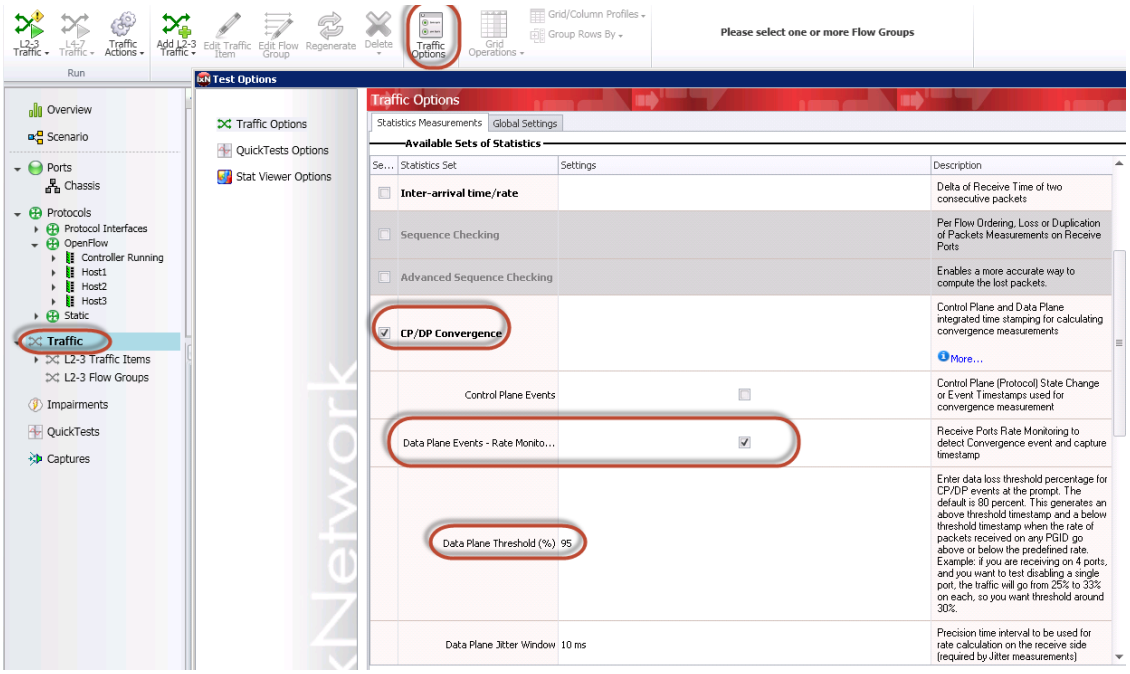


Figure 56: Traffic Options window

13. Go to **Traffic Wizard** window to create traffic flow between Host-1 to Host-2 and Host-3.
    a. Select **Traffic** from the tree and click on **Add L2-3 Traffic** button in the ribbon. It will open **Advance Traffic Wizard**.

Figure 57: Advanced Traffic Wizard

    b. Select **Type of Traffic** as IPv4and use **OpenFlow** encapsulation filter for **Source** and **Destination** endpoints.

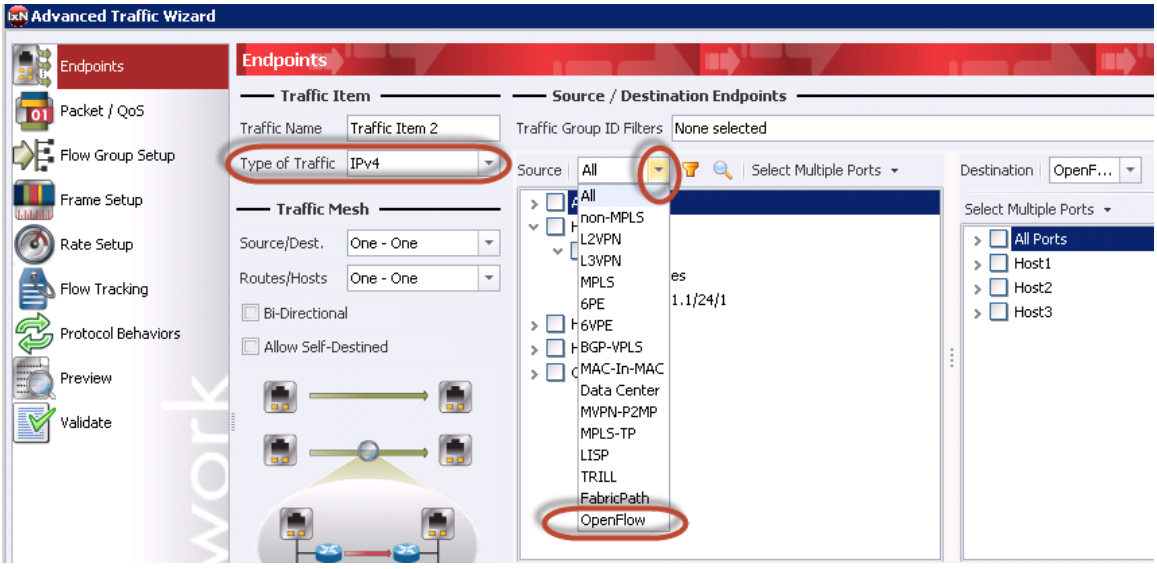


Figure 58: Advanced Traffic Wizard, Endpoints window

c. Select **Host-1** as traffic source and **Host-2** and **Host-3** as traffic destination. Make sure the **Merge Destination Range** check box is selected. This ensures that unique flow is created based on destination IP address.

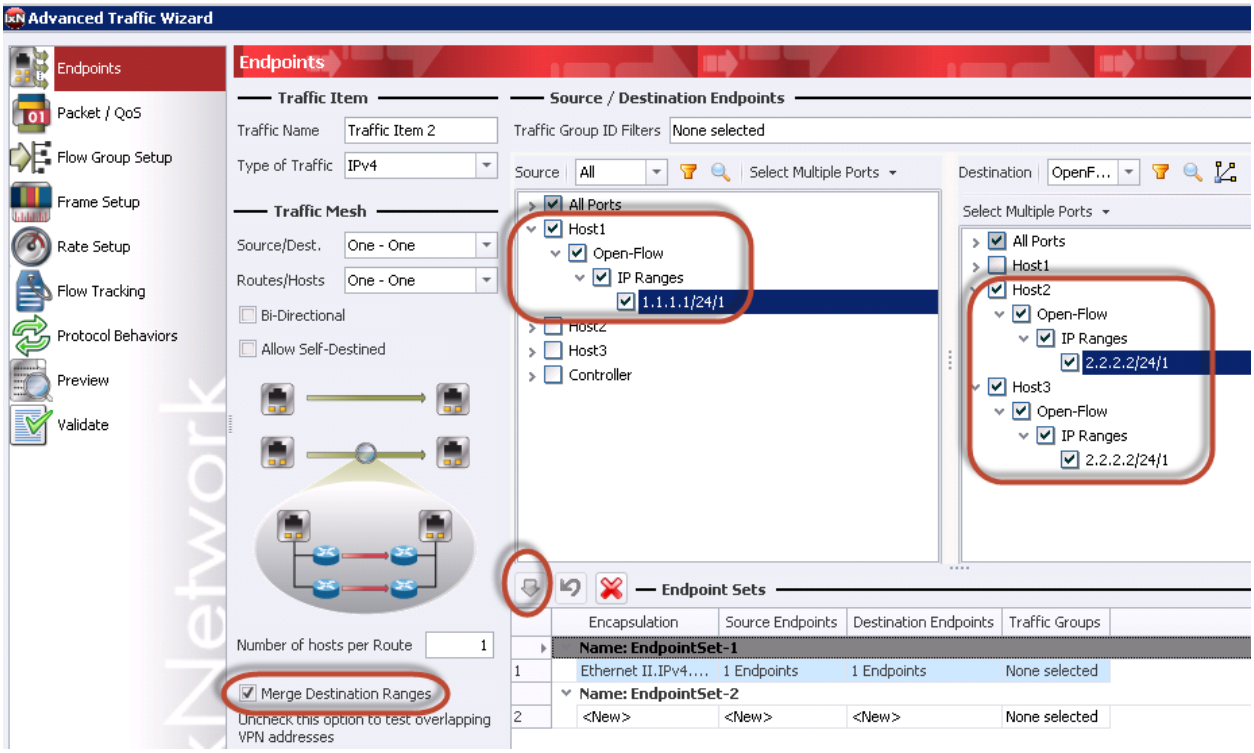


Figure 59: Advanced Traffic Wizard, Source/Destination Endpoints mapping

d. Leave default values on Packet/QoS, Flow group setup, and frame setup windows.

e. Set the desired traffic load on **Rate Setup** window and click **Next**.

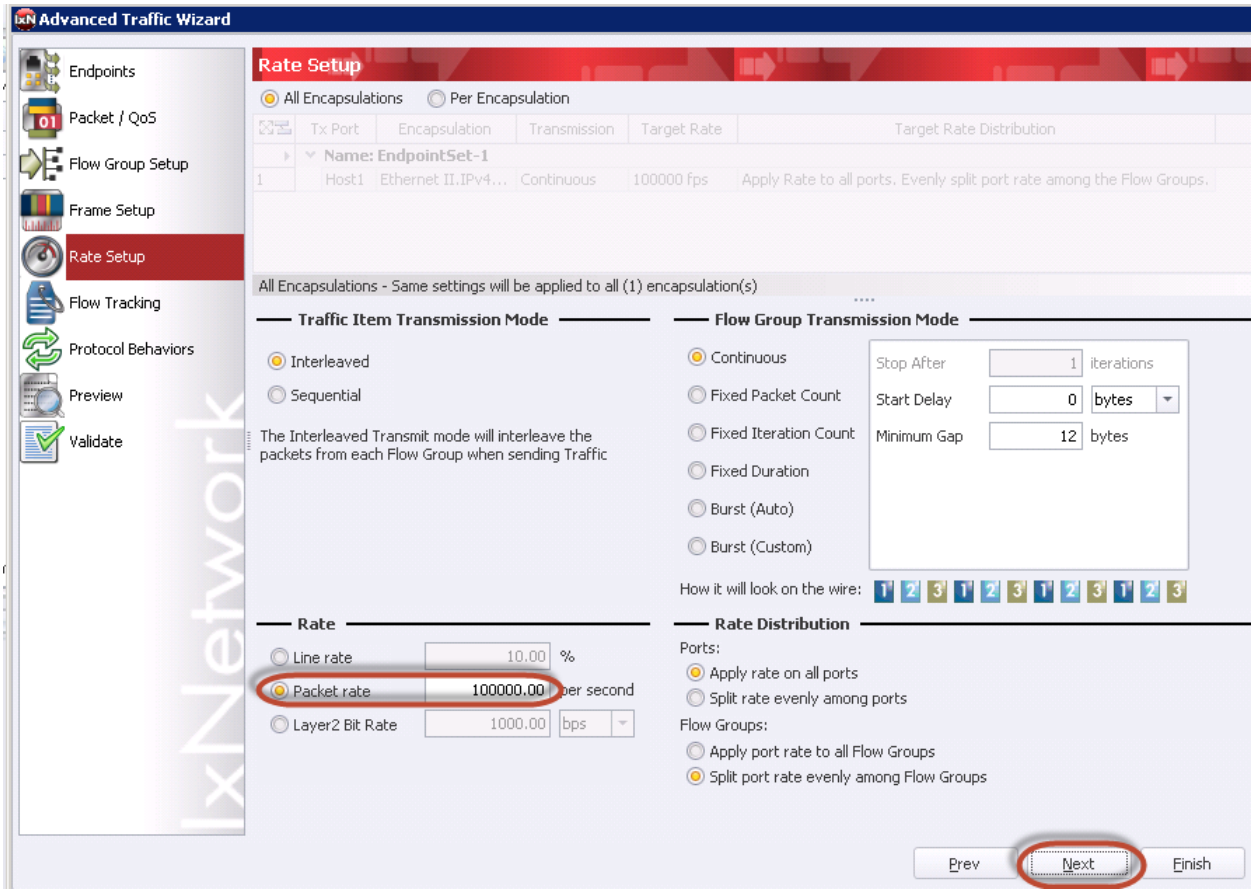


Figure 60: Advanced Traffic Wizard, Rate Setup window

f. On **Flow Tracking** window, select **Source/Dest Value Pair** and **Dest Endpoint** as tracking option. Click **Next**.

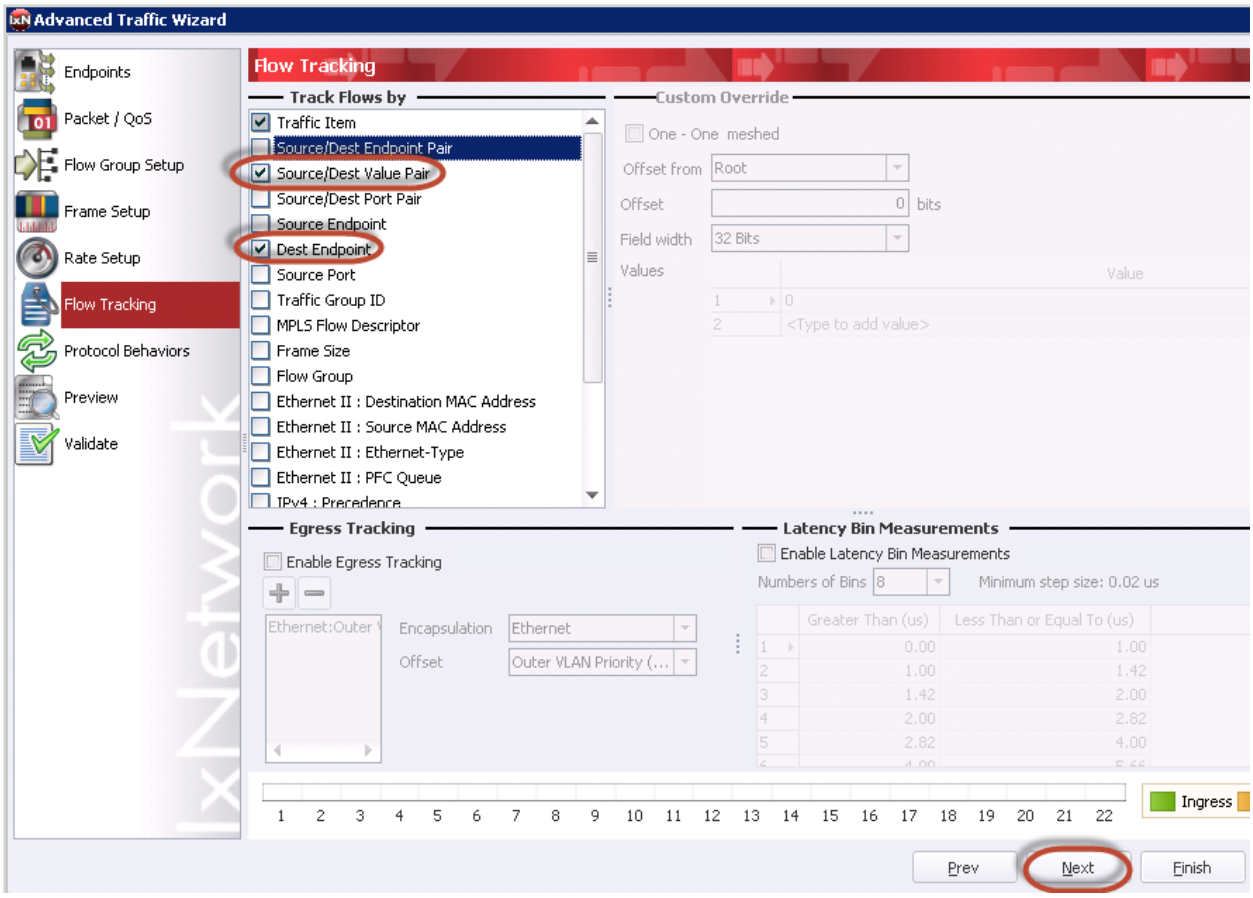


Figure 61: Advanced Traffic Wizard, Flow Tracking window

g. Skip **Protocol Behaviors** window and go to **Preview** window to view how the traffic flow looks like and click **Finish** button.

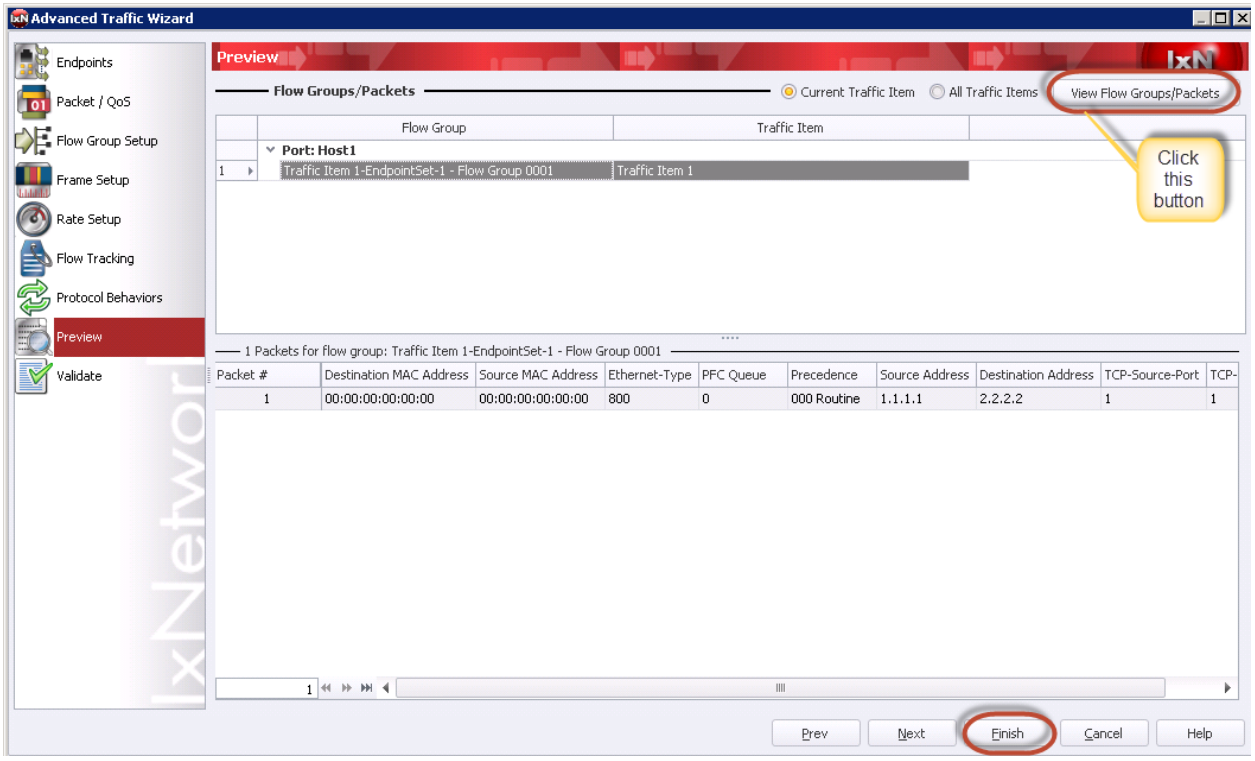
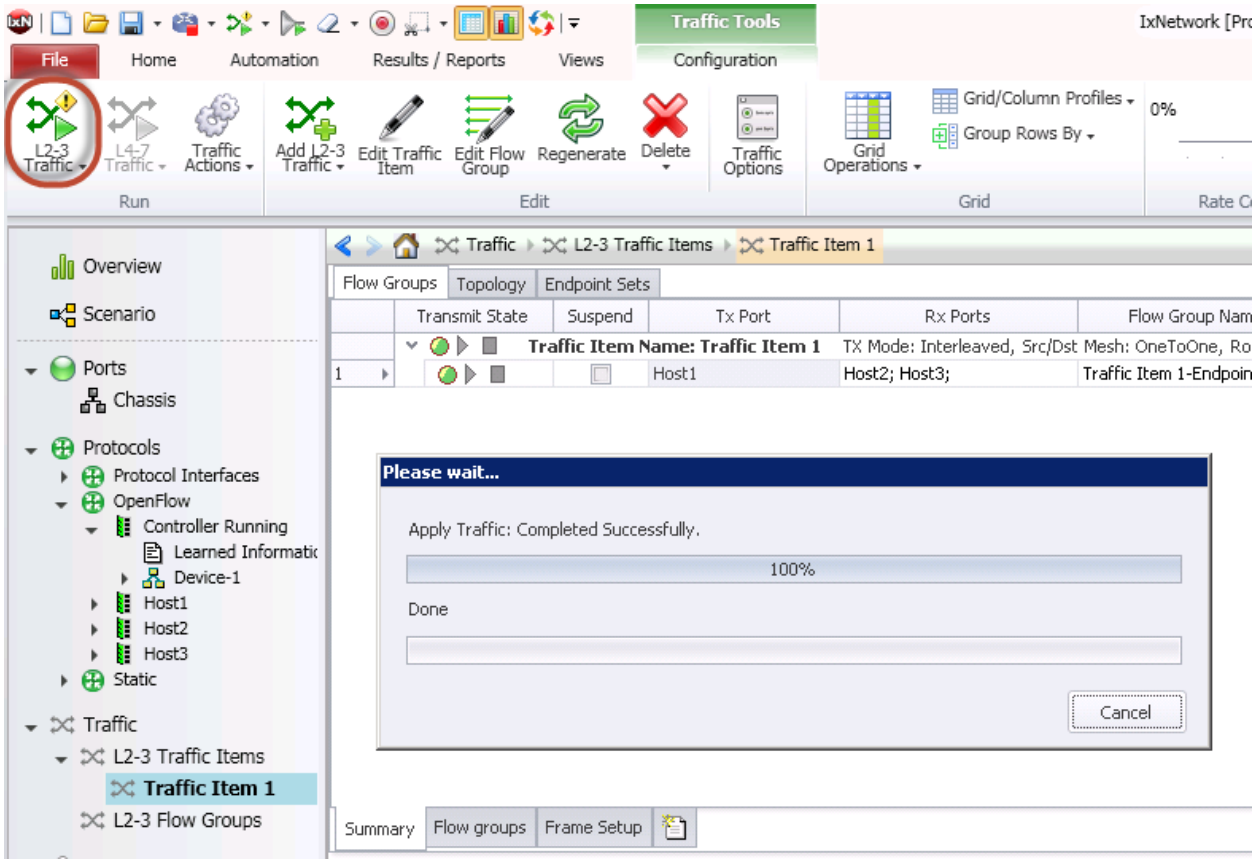


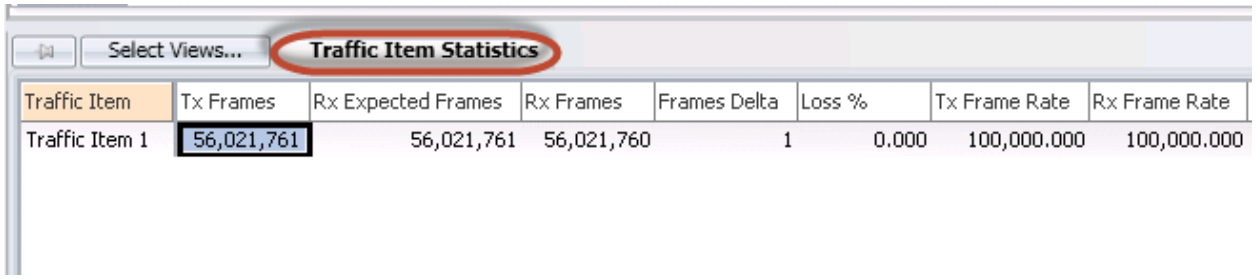
Figure 62: Advanced Traffic Wizard, Preview window

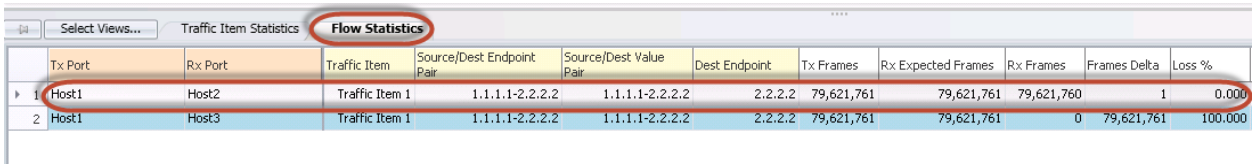
14. Click **L2-3 Traffic** button to push the traffic on port and start traffic.



Result Analysis

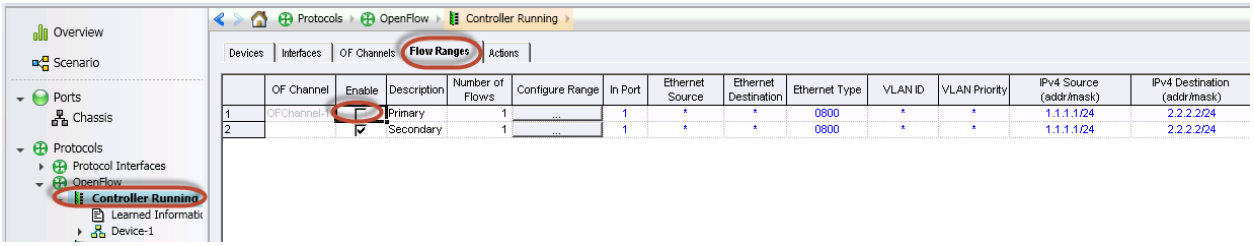
1. Verify that there is no traffic loss and traffic is flowing through the primary path (which has highest priority, as configured earlier) on the switch.



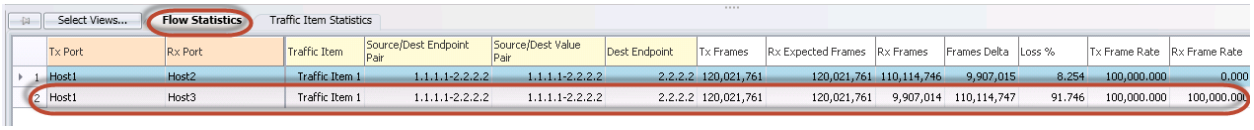


2. While traffic is running, go to OpenFlow controller flow range and **disable Primary** flow. This sends Flow-Mod delete command to switch. Switch removes the Primary Flow entry

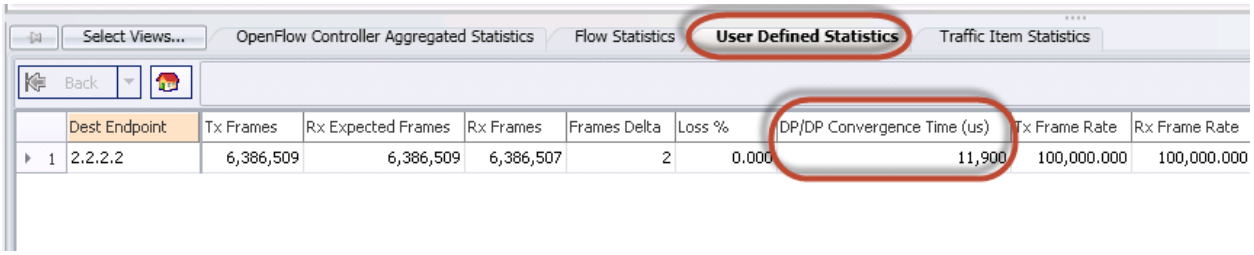
and since there is only one flow entry in its Flow Table it will start forwarding packets out on the secondary path output port.



3. Go to traffic **Flow Statistics** window and verify that the traffic gets switched over to secondary path towards host-3. And, go to **Traffic Item Statistics** tab to verify there is no traffic loss.

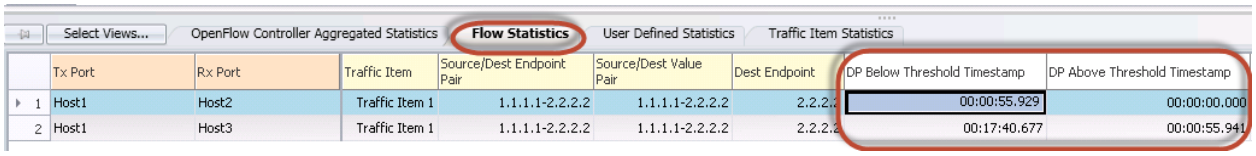


4. From traffic item statistics view, perform drill down per destination endpoint (right click on traffic item, select drill down per destination option). This will create User Defined Statistics view tab. On this tab, you will notice **DP/DP Convergence Time** on far right hand side (you can move this column in the beginning by dragging the column header). This field provides data-plane event convergence time in micro seconds.

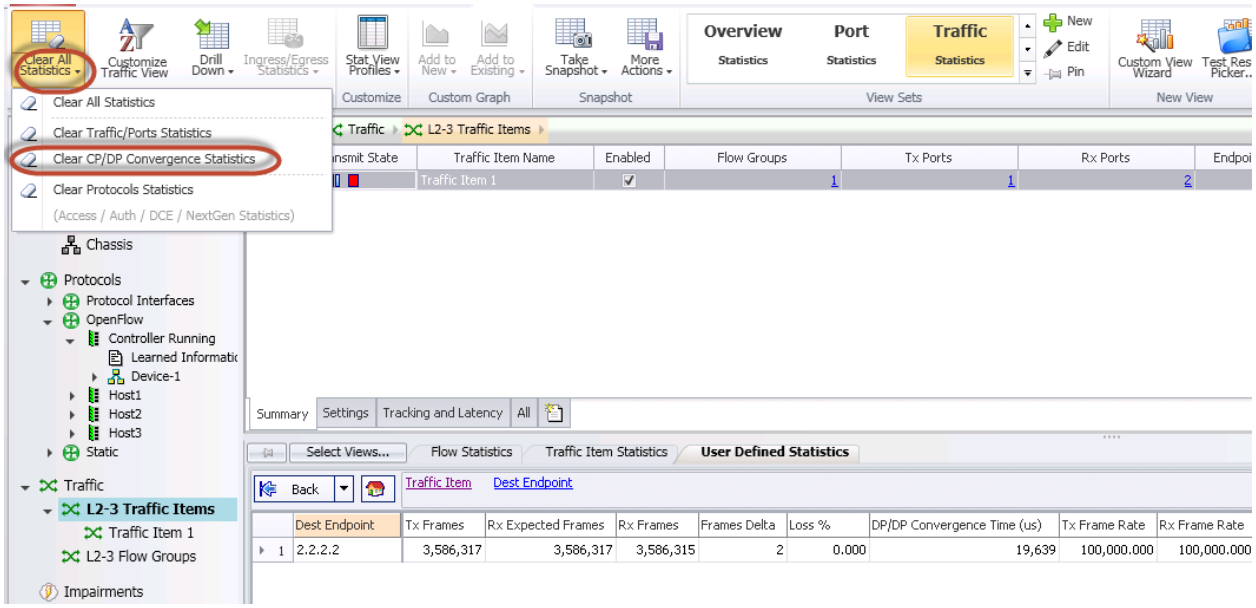


5. Go to **Flow Statistics** to monitor DP-Above and DP-Below timestamp (located in far right corner) to confirm DP/DP convergence time accuracy.

DP-DP Convergence time = tDP-Above time stamp – tDP-Below timestamp



6. Prior to next control-plane trigger event, perform **Clear CP/DP Convergence Statistics** to get accurate results for the next convergence event on the switch.



Conclusions

This test allows you to validate DUT's ability to detect the failure and switch traffic to alternate path as fast as possible without significant packet loss. It also provides accurate traffic convergence measurements.

Variables

1. TrueView convergence test also provides accurate measurement for control-plane to data-plane convergence time. To perform this test, user should select control-plane events check box in step#17. All other steps remain the same.

   Go to Statistics view and track by destination endpoint, you should notice CP/DP convergence Time column.

2. Same test can be repeated for various traffic rates, number of flows and threshold values.

Test case: OpenFlow Controller Scalability Test

Overview

SDN continues to gain momentum in the networking industry. The OpenFlow protocol is widely accepted and leading the SDN trend. Almost all major network equipment manufacturers as well as startups have shown interest in OpenFlow and developing OpenFlow Controller, OpenFlow Switch, or both in their product portfolio.

Ixia continues to develop OpenFlow protocol emulation. Ixia has introduced v1.0 Controller emulation in 2012 and now is introducing v1.0 Switch emulation. Using OpenFlow switch emulation, one can validate basic functionality of the OpenFlow controller. Also, OpenFlow switch emulation helps in performance measurement and scalability.

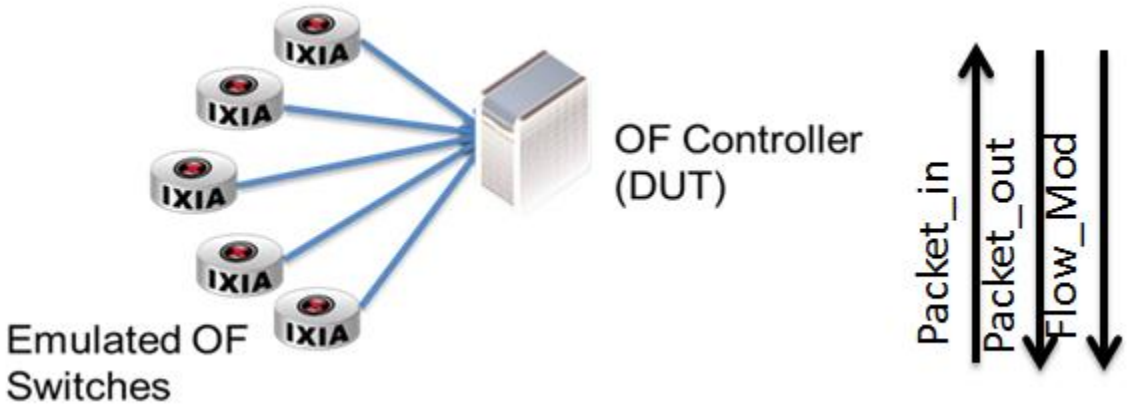
Objective

The objective of this test case is to verify functionality and scalability of the OpenFlow controller by performing following tasks:

- Create multiple switch emulation and bring up OF channel using single Ixia port,
- Push thousands of flows with different match/actions to ensure that controller can push the flows correctly.
- Generate various messages to the controller, such as packet_ins, port status message, vendor message, and error message to ensure that controller is capable to respond each message timely and accurately.

For this test case, use 2 Ixia ports connected in back-to-back mode. One port is emulating OpenFlow controller and the other port is emulating multiple OpenFlow switches.

Setup

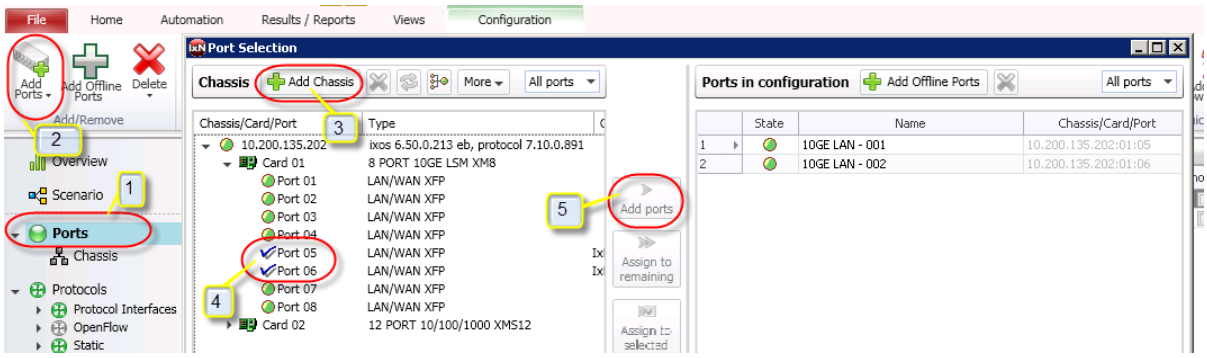


Step-by-Step Instructions

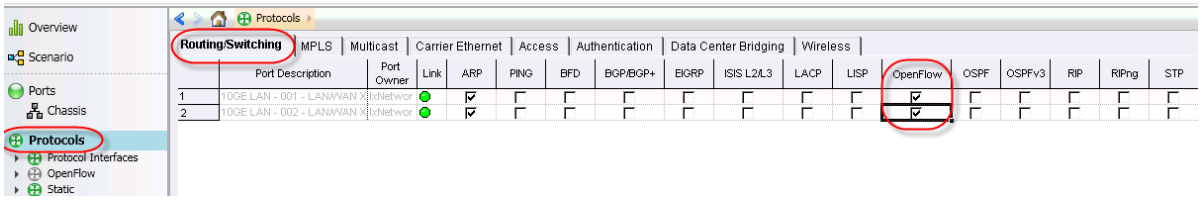
The following steps describe the procedure for performing the test.

1. Reserve 1 Ixia port

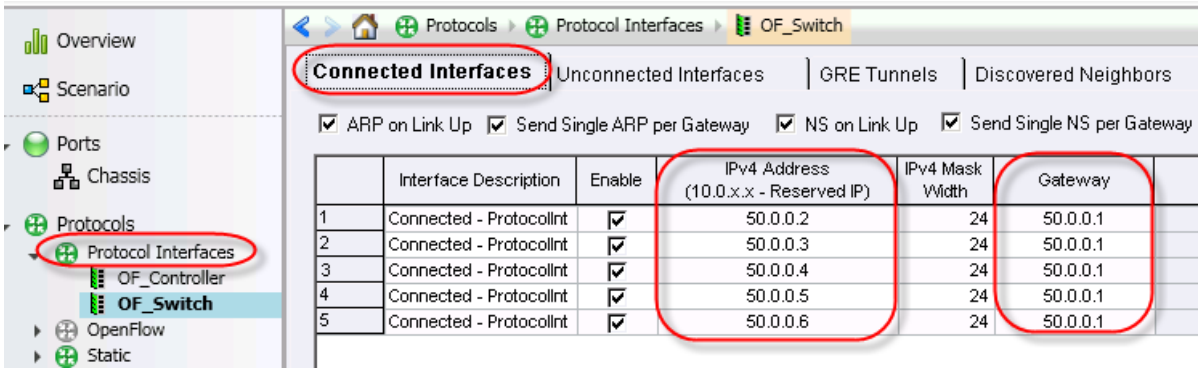
   Note: This example uses 2 Ixia ports connected in back-to-back mode. One port is emulating OF Controller and other port is emulating OF Switch



2. In the **Protocols** Window, select **OpenFlow** checkbox to enable the OpenFlow.

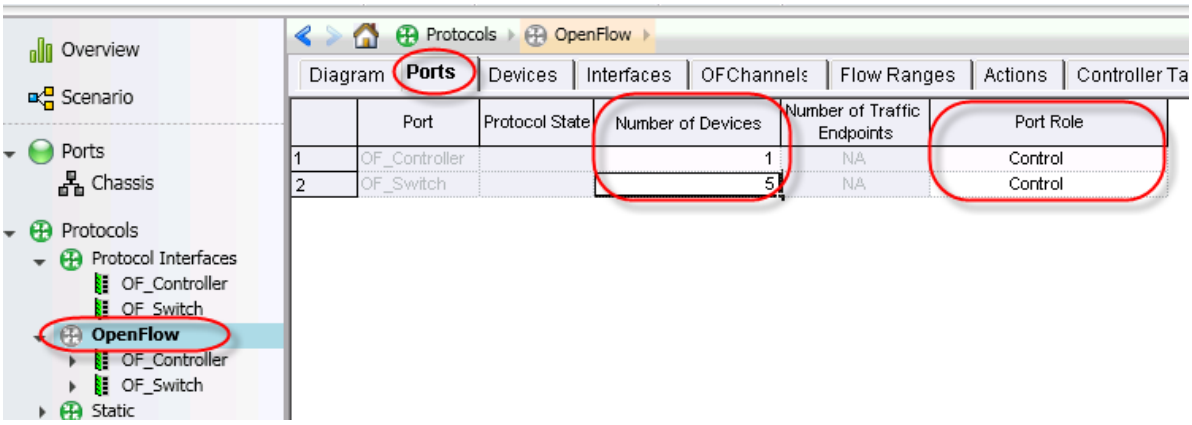


3. Create 5 protocol interfaces (to emulate 5 OF Switches). Configure IP addresses of emulated OF Switches and **Gateway** address from the **Connected Interface** tab in the **Protocol Interfaces** window. For OF Channel, ensure that ARP is resolved.

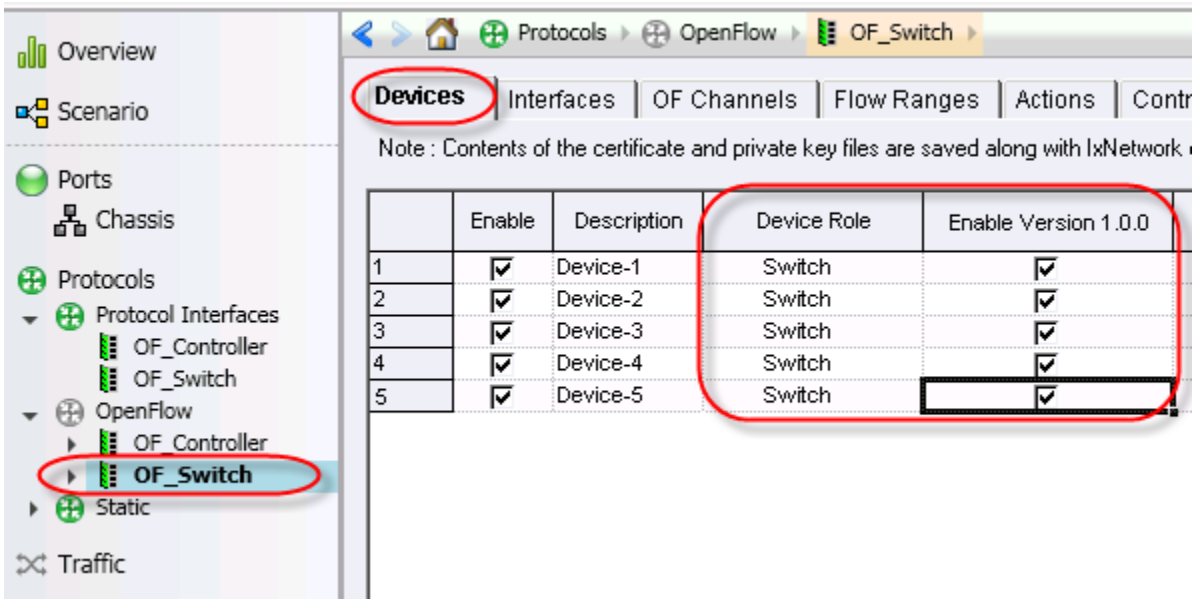


4. Define the port role by selecting the role as **Control** from the **Port Role** list in the **Ports** tab in the **OpenFlow** window.

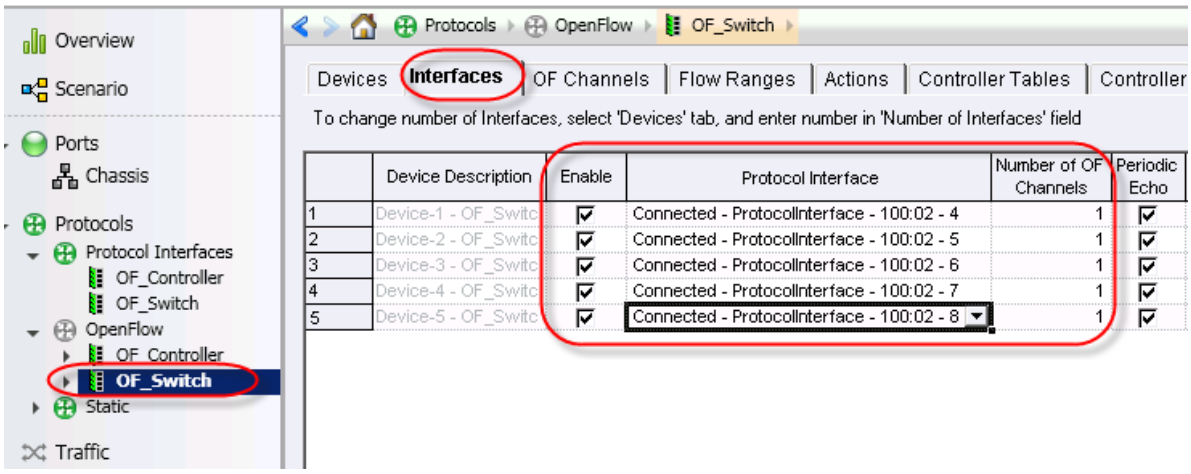
5. Define **Number of Devices** as 5 in the **Ports** tab.



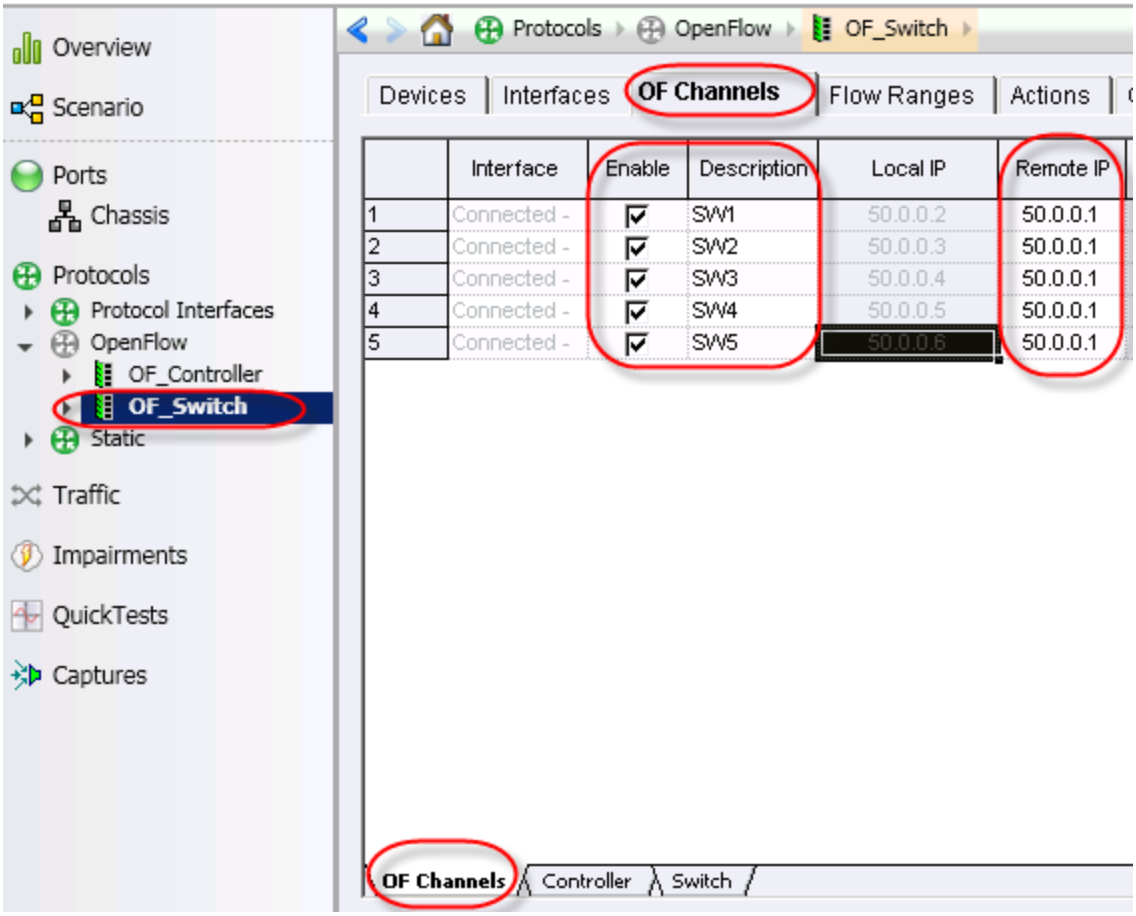
6. Click the **Devices** tab. Modify the **Device Role** to **Switch**. Also, select **Enable Version 1.0.0** checkbox.



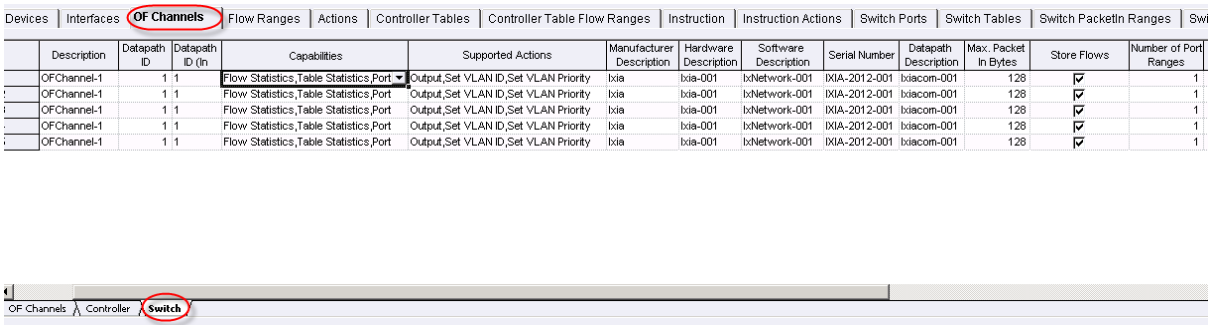
7. Click the **Interfaces** tab. Assign the **Protocol Interfaces** that you created in the **Protocol Interface** window. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1.* Leave all other parameters as default.



8. Click **OF Channels** tab. Enter the IP address of OpenFlow Controller for **Remote IP** and select the **Enable** checkbox. Also change switch name in the **Description** column.

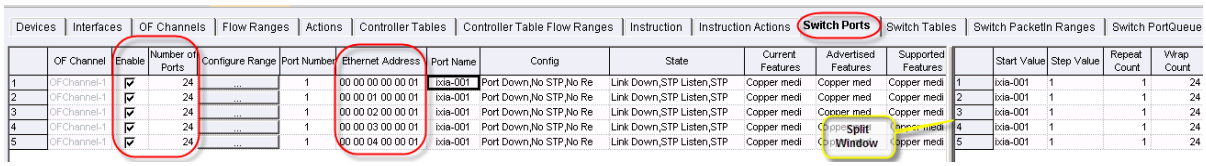


9. In the **OF Channels – Switch** tab, change **Number of Table Ranges** to 4 and leave all parameter as default. However, if it is required to test OF Controller with different capabilities, datapath ID, supported actions, and so on, then those parameters can be changed from this tab.



10. Click **Switch Ports** tab, and enable the **port range**. Change **Number of ports** to 24, **Ethernet Address** for each switch. Other parameters, such as port state, configuration, supported/advertised features, and so on, can be changed from this section.

    Note: Configure Range or Split window can be used to define start/step value to create uniqueness



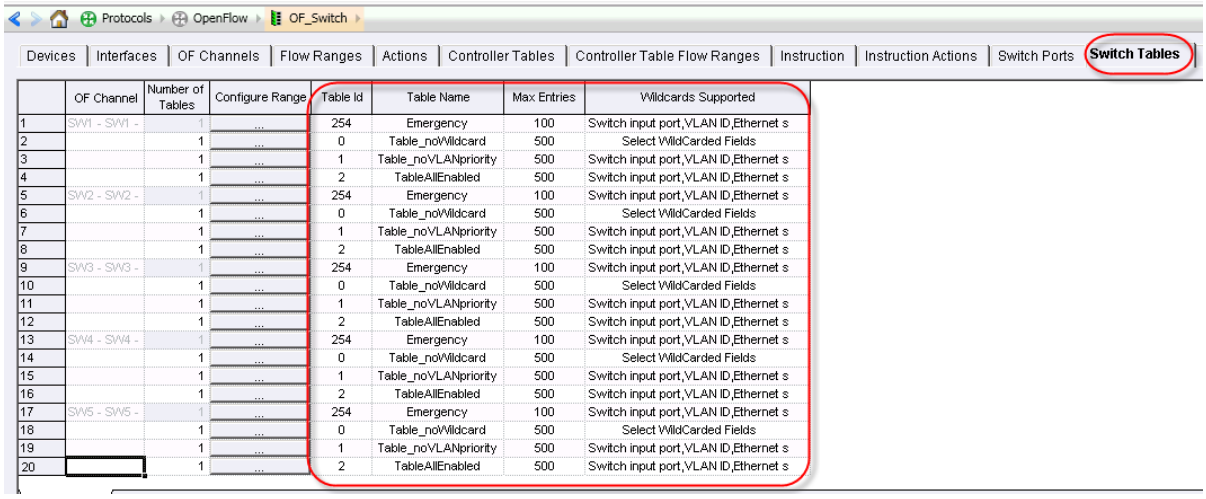
11. In the **Switch Tables** tab, make following changes:

- **Table ID –** Define unique table IDs for each switch
- **Table Name (optional) –** Give meaningful name
- **Max Entries –** Desired number of flows that each table can hold
- **Wild Card Supported –** Enable/Disable checkbox for any field that requires wildcard support. For example, if you disable wildcard support for VLAN ID and if switch receives any flow that contains wildcard for VLAN ID field, then it does not install the flow in that table

The following example (below snapshot) shows that each switch has 4 tables. All 5 switches have same tables.

It is defined as:

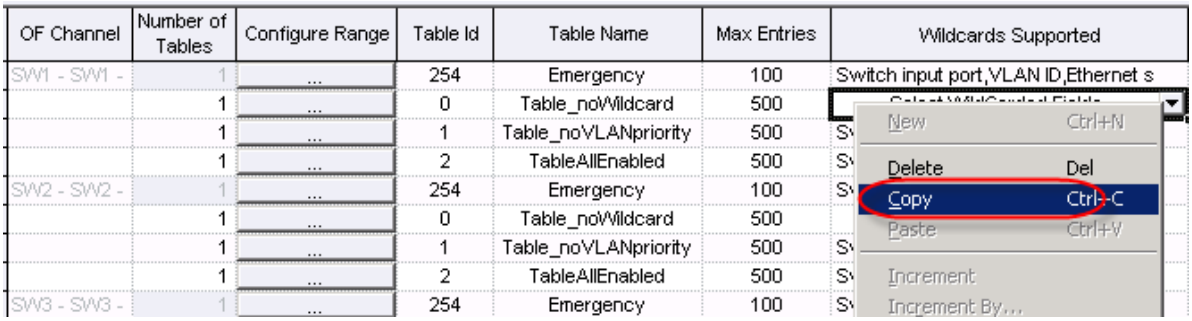
- **Table 1 –** Emergency, Switch uses these flows if the OF channel connection to the controller gets reset or lost
- **Table 2 –** Table with no wildcard support, it means this table does not accept any flows that have wildcard character in it.
- **Table 3 –** Table with no wildcard support for VLAN priority, this table does not accept any flows that has wildcard set for VLAN priority field
- **Table 4 –** Table with all wildcard, this table accepts any flows that has wildcard or no wildcard



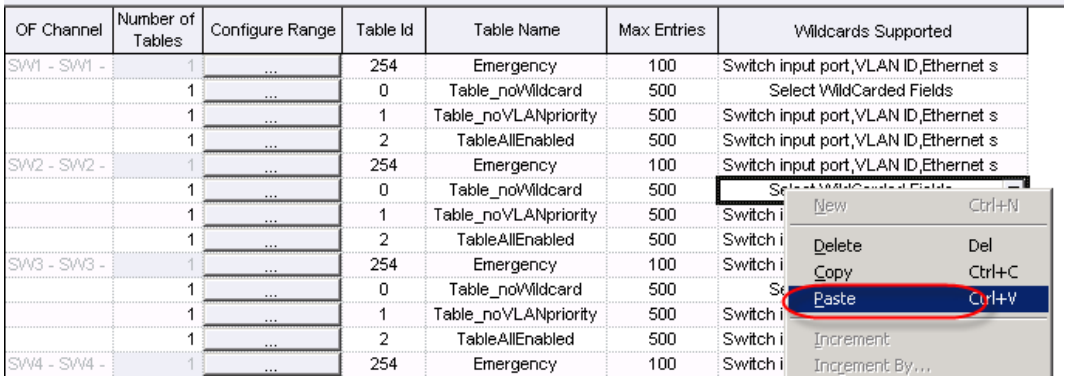
	OF Channel	Number of Tables	Configure Range	Table Id	Table Name	Max Entries	Wildcards Supported
1	SW1 - SW1 -	1	...	254	Emergency	100	Switch input port,VLAN ID,Ethernet s
2		1	...	0	Table_noWildcard	500	Select WildCarded Fields
3		1	...	1	Table_noVLANpriority	500	Switch input port,VLAN ID,Ethernet s
4		1	...	2	TableAllEnabled	500	Switch input port,VLAN ID,Ethernet s
5	SW2 - SW2 -	1	...	254	Emergency	100	Switch input port,VLAN ID,Ethernet s
6		1	...	0	Table_noWildcard	500	Select WildCarded Fields
7		1	...	1	Table_noVLANpriority	500	Switch input port,VLAN ID,Ethernet s
8		1	...	2	TableAllEnabled	500	Switch input port,VLAN ID,Ethernet s
9	SW3 - SW3 -	1	...	254	Emergency	100	Switch input port,VLAN ID,Ethernet s
10		1	...	0	Table_noWildcard	500	Select WildCarded Fields
11		1	...	1	Table_noVLANpriority	500	Switch input port,VLAN ID,Ethernet s
12		1	...	2	TableAllEnabled	500	Switch input port,VLAN ID,Ethernet s
13	SW4 - SW4 -	1	...	254	Emergency	100	Switch input port,VLAN ID,Ethernet s
14		1	...	0	Table_noWildcard	500	Select WildCarded Fields
15		1	...	1	Table_noVLANpriority	500	Switch input port,VLAN ID,Ethernet s
16		1	...	2	TableAllEnabled	500	Switch input port,VLAN ID,Ethernet s
17	SW5 - SW5 -	1	...	254	Emergency	100	Switch input port,VLAN ID,Ethernet s
18		1	...	0	Table_noWildcard	500	Select WildCarded Fields
19		1	...	1	Table_noVLANpriority	500	Switch input port,VLAN ID,Ethernet s
20		1	...	2	TableAllEnabled	500	Switch input port,VLAN ID,Ethernet s

Note: Use Copy/Paste functionality to create same table on each switch.

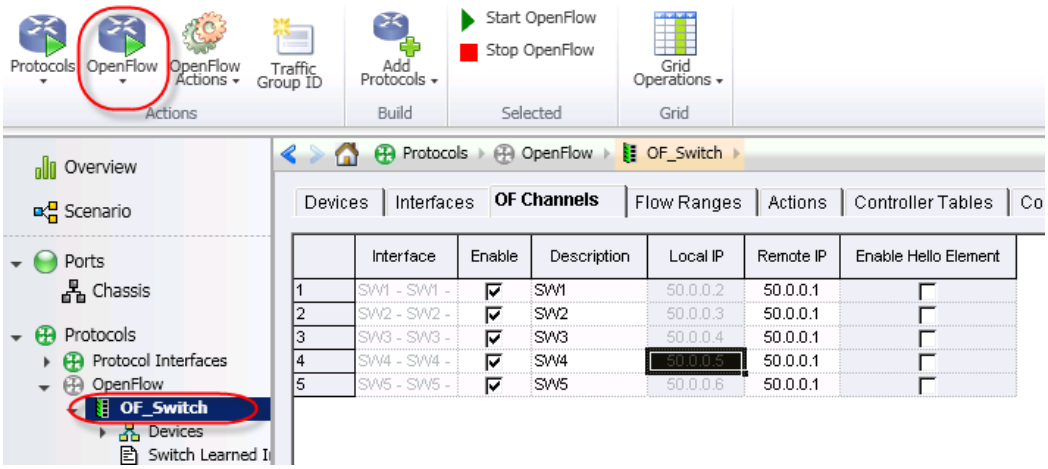
Select the table > Right click > **Copy**



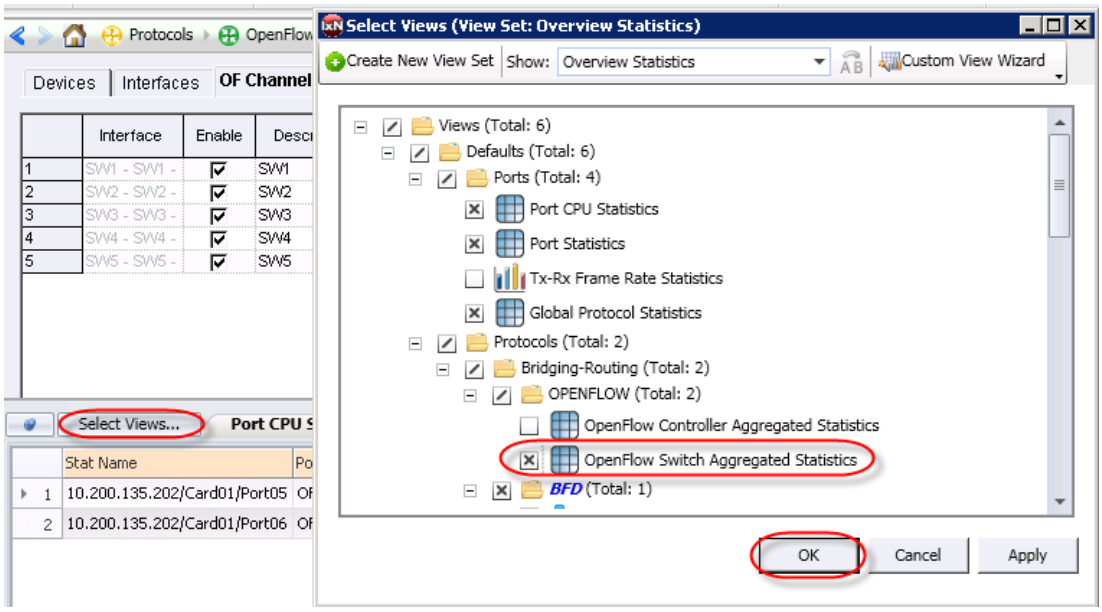
For paste, select the table > Right click > **Paste**



12. Start **OpenFlow** protocol.

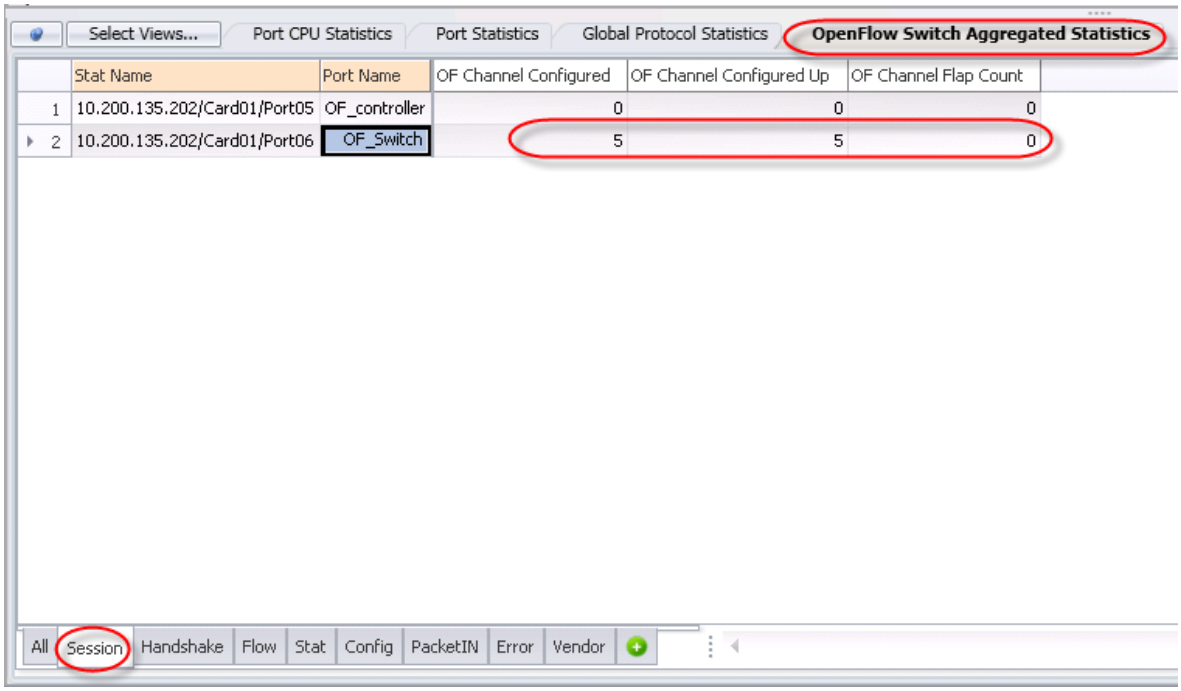


13. Enable **OpenFlow Switch Aggregated Statistics** view.

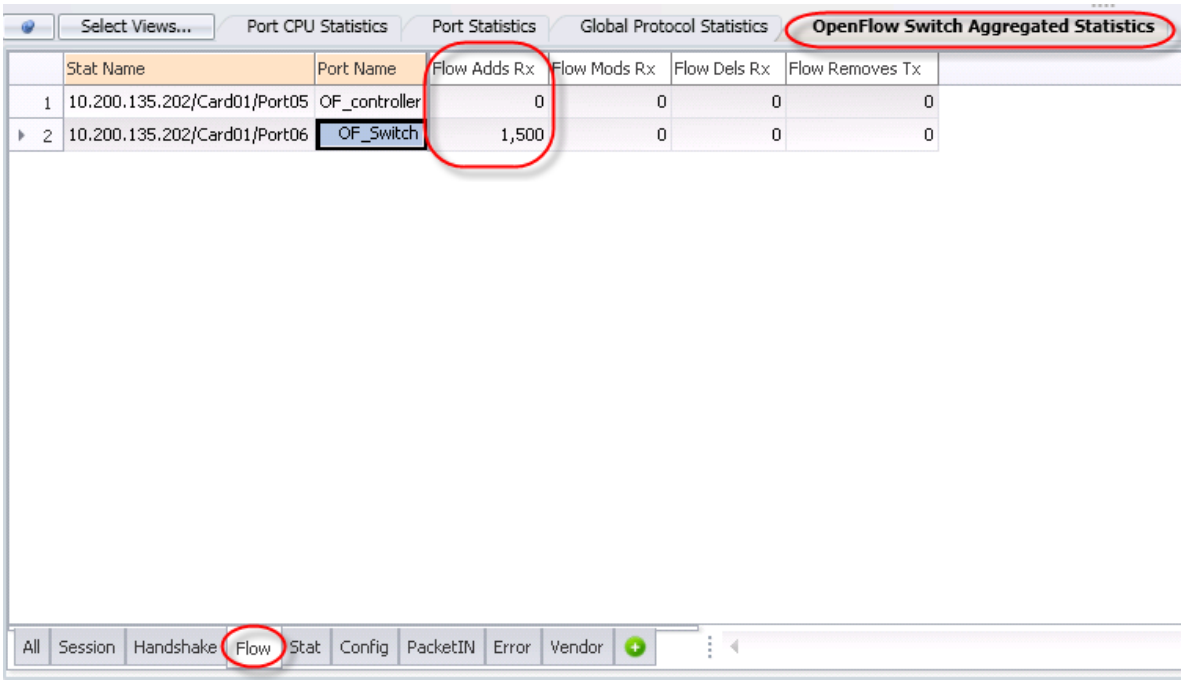


14. Click the pre-defined filter tabs at the bottom of the window to observe statistics like **Sessions, Flow,** and **Error.**

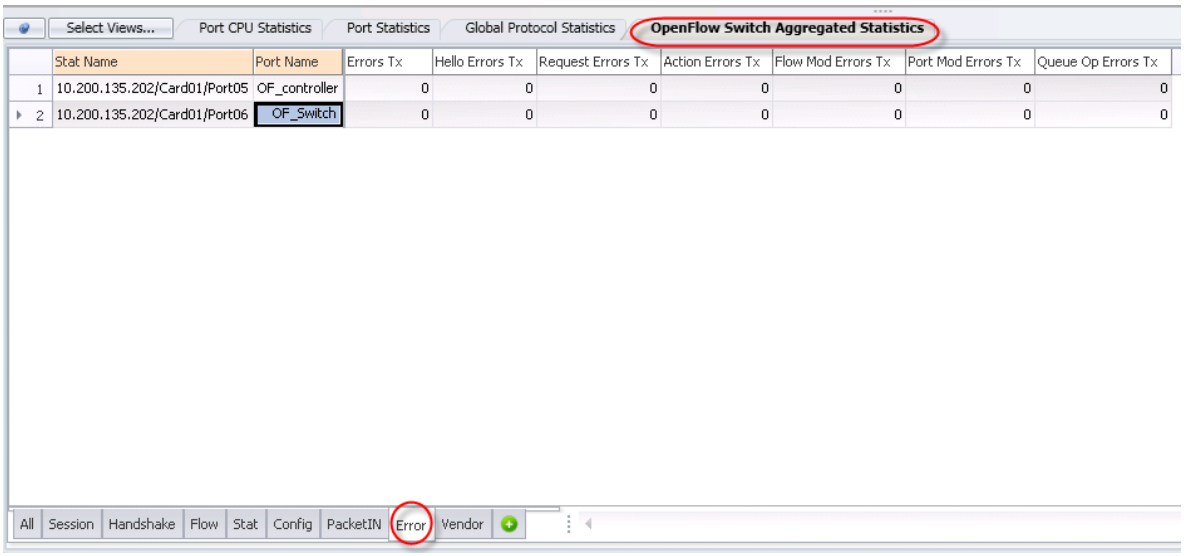
Sessions:



Flow:

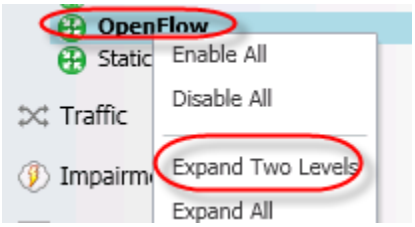


Error:

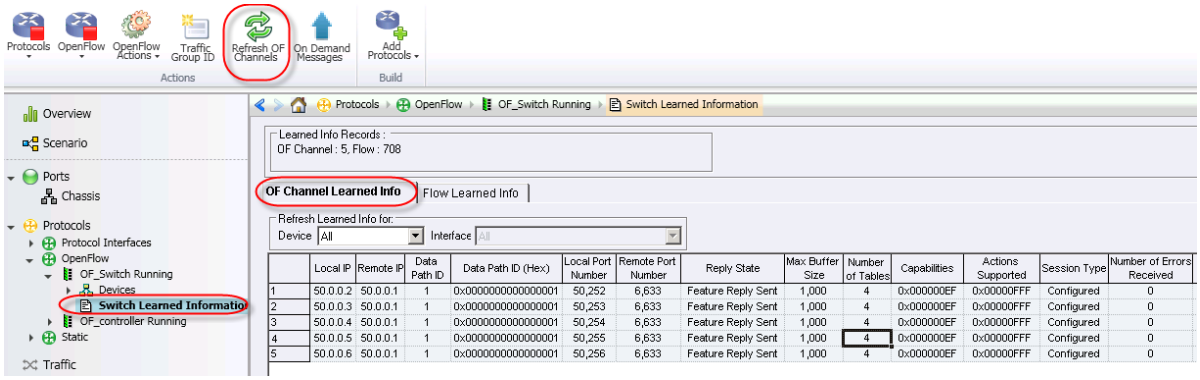


15. Expand **OpenFlow Switch view** and select **Switch Learned Information.**

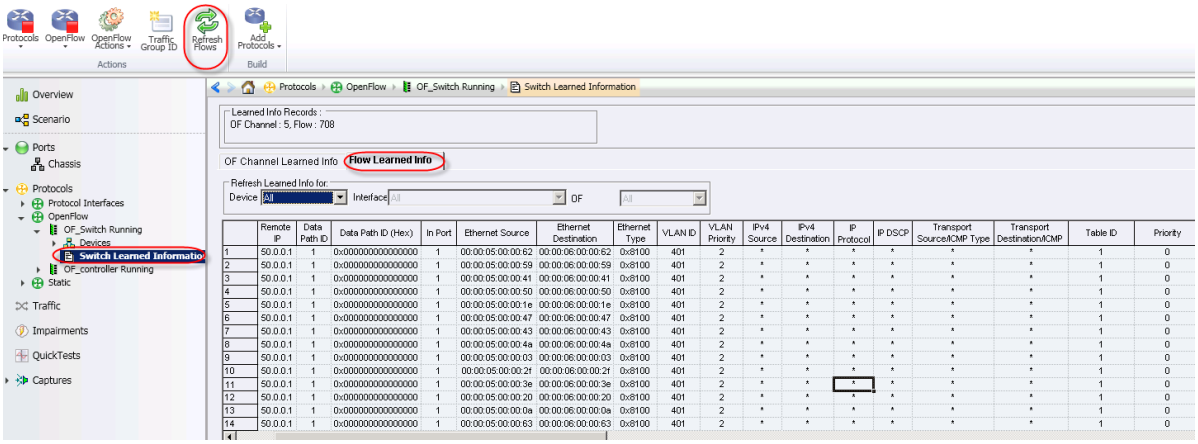
    Note: If your view is in summary mode (which is default), you can select OpenFlow protocol and right click then select **Expand Two Levels** option



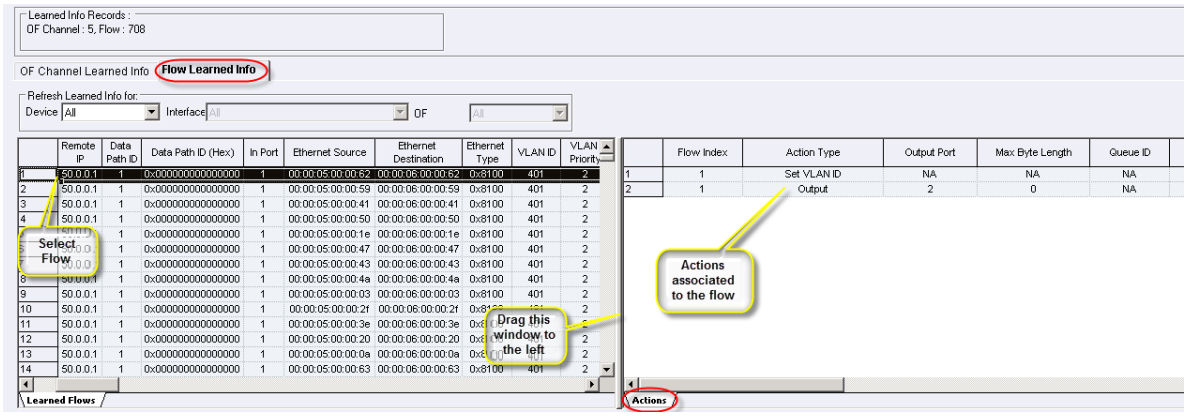
16. In the **Switch Learned Information** tab, click **Refresh OF Channels** button.
    This action generates Feature Reply message to the controller. This message displays useful information like, number of OF Channels, number of tables, capabilities, supported actions, and number of errors.



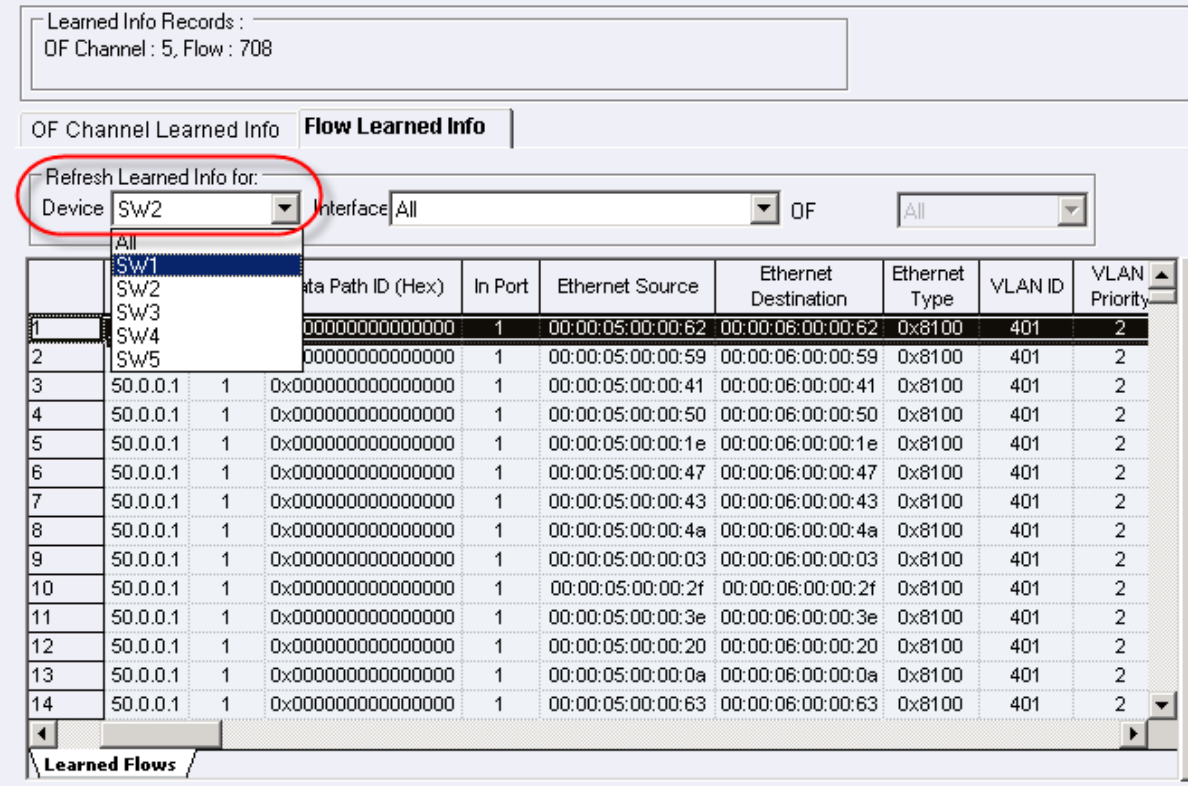
17. Click **Flow Learned Info** tab, and then click **Refresh Flows** button. It displays flows that were pushed by the controller. It also displays information about the **Match** field.



18. To see list of **ACTIONS** associated to the flow, reduce the left window pane size, select the desired flow, and observe actions in the split window.



19. Use the **Device filter** to see learned flow information for a particular device.



Conclusions

This test case validates OpenFlow Controller's functionality and scalability by establishing multiple OpenFlow channels (TCP or secure TLS) with the emulated switches. Also, it can push thousands of flows with various match/actions and validate its accuracy from the learned information. In addition to that from the switch, you can generate other messages, such as Packet_in, port status message, and various error messages like flow_table_full, bad stat requests to ensure controller can handle these in coming message appropriately.

Variables

Use the following variables to test controller's scalability:

- Add large number of OpenFlow switches with different set of capabilities, configuration, and supported features/actions
- Generate various packet_in profiles and generate packet_in messages to the controller to validate the responsiveness of the controller

Test case: Packet_out Rate Calculation

Overview

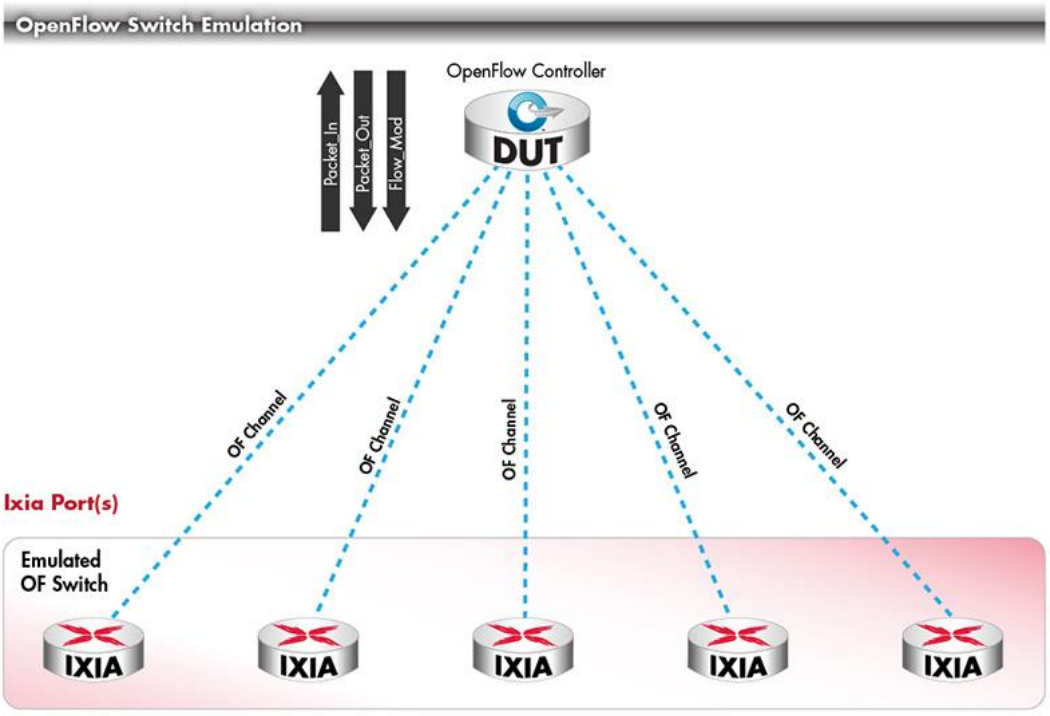
In the event, if OpenFlow switch receives any packets that do not have matching flow entry in its table, then it normally consults OpenFlow controller by generating packet_in message to the controller and let controller make the decision. Therefore, it is very important to test controller's responsiveness and make sure it learns and builds correct flow table for each switch as per the configured policy.

If the controller is not able to handle incoming packet_in message in timely manner, then it causes packet drops, higher latency, and network disruption.

Objective

This test case measures controller's responsiveness and accuracy of packet_in handling. Using packet in range you can generate various types of packet_in messages and measure the packet_out/flow_mod response time.

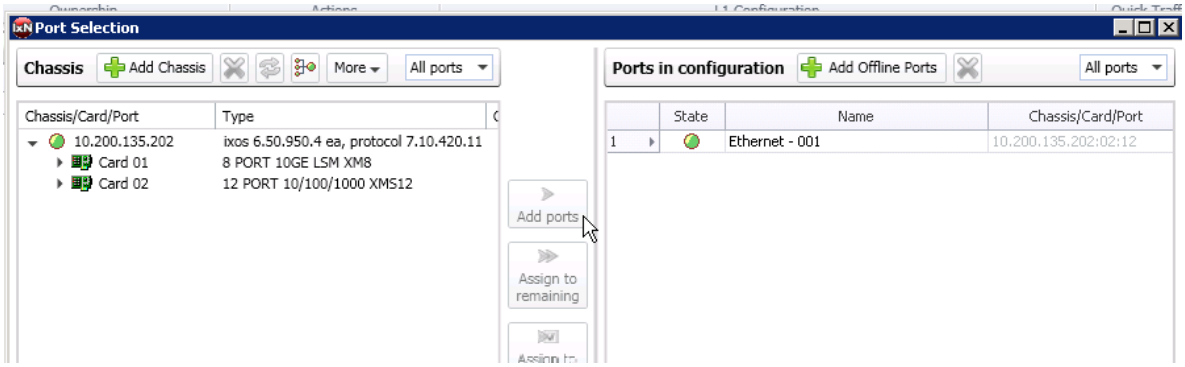
Setup



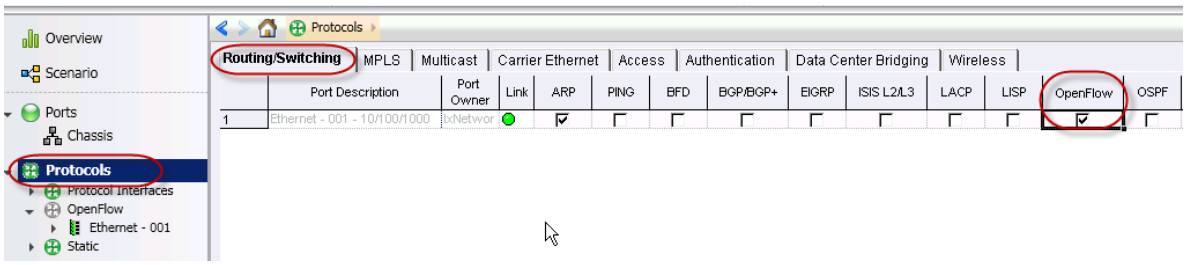
Step-by-Step Instructions

The following steps describe the procedure for performing the test.

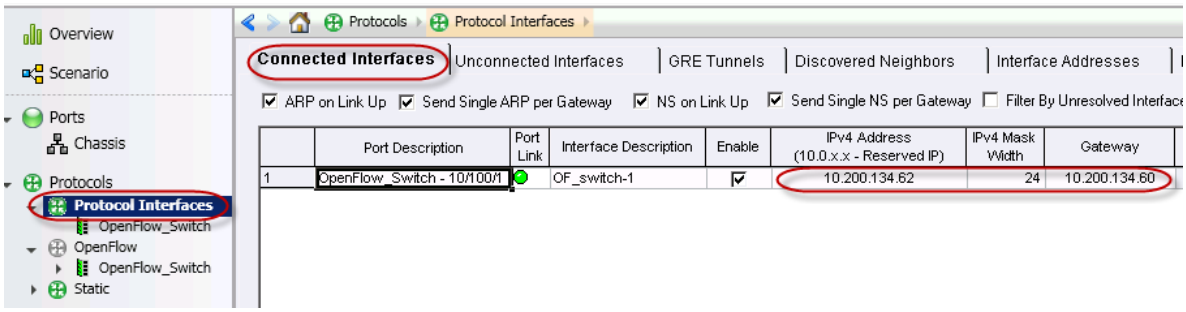
1. Reserve 1 Ixia port for OpenFlow switch emulation.



2. In the **Protocols** window, select **OpenFlow** checkbox to enable OpenFlow on port.

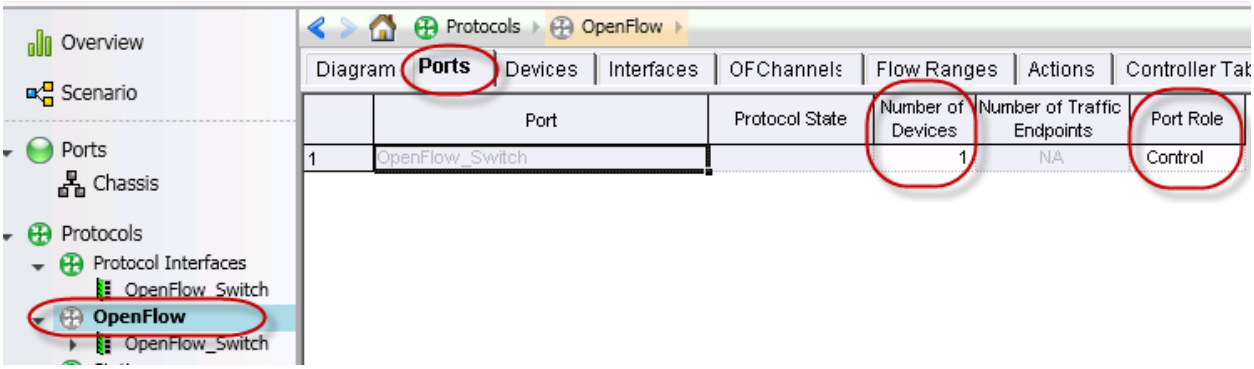


3. Click the **Connected Interface** tab to configure the emulated OpenFlow switch, IP address, and **Gateway** address in the **Protocol Interfaces** window. For OF Channel, ensure that ARP is resolved.

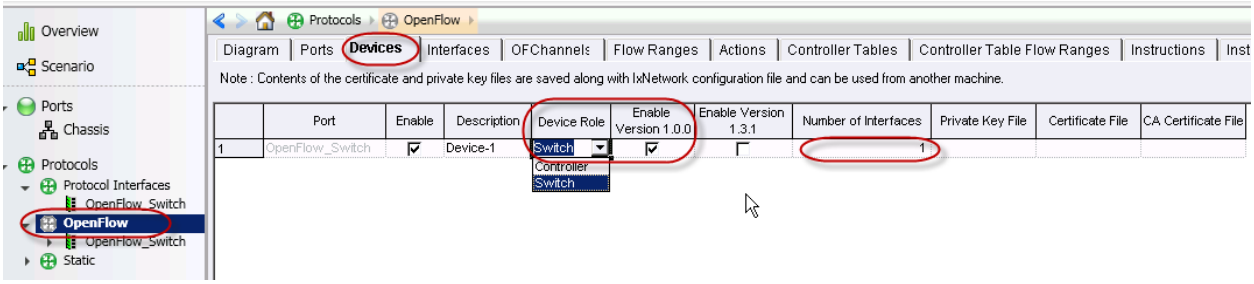


4. Define the **Number of Devices** and the port role as **control** by selecting the role from the **Port Role** list on the **Ports** tab on the **OpenFlow** window.

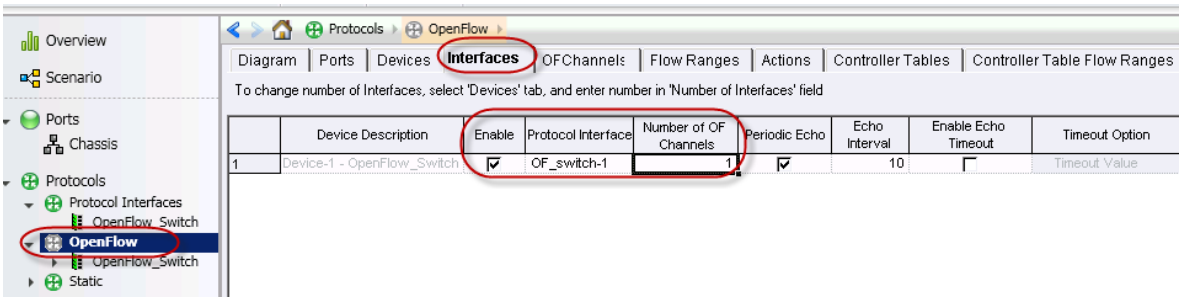
   Note: Number of Devices option allows creating multiple OpenFlow Switch emulation on a single physical Ixia port. Make sure, unique protocol interface is created and assigned to each emulated switch.



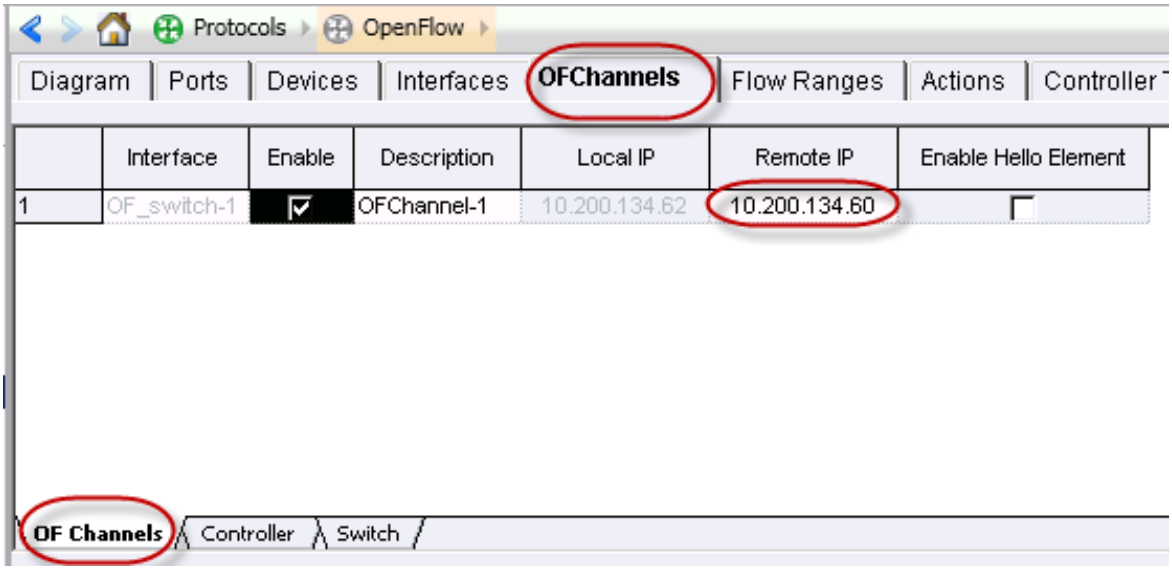
5. Click the **Devices** tab and configure following parameters:
   - **Device Role** as Switch
   - **Enable version** v1.0.0 (v1.3.1 is not currently supported)
   - **Number of Interfaces** as 1
   - If secured TLS OF channel connection is desired then specify key file path **(Optional)**



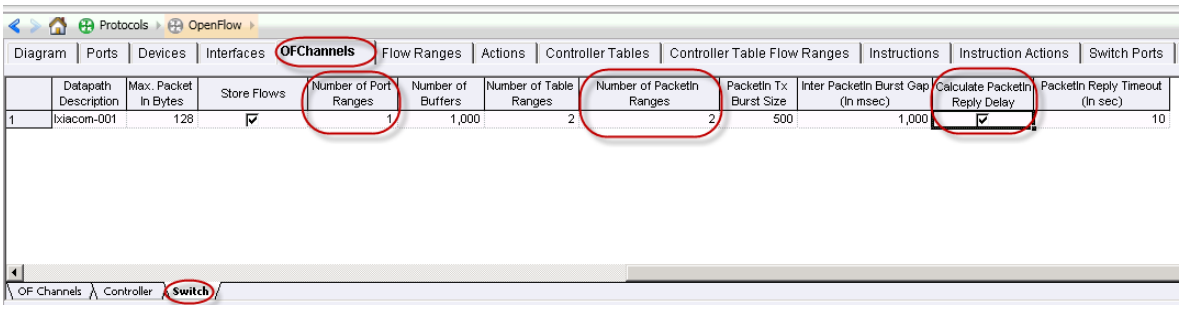
6. Click the **Interface** tab of the **OpenFlow** window and assign the **Protocol Interfaces** that you created in the **Protocol Interface** window. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1*



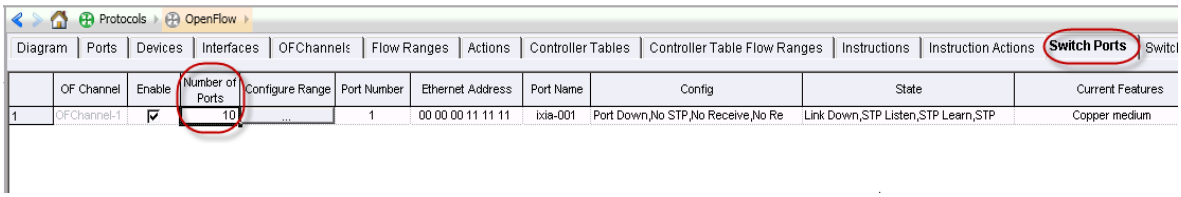
7. In the **OF Channels** tab specify the controller's IP address in **Remote IP** field.



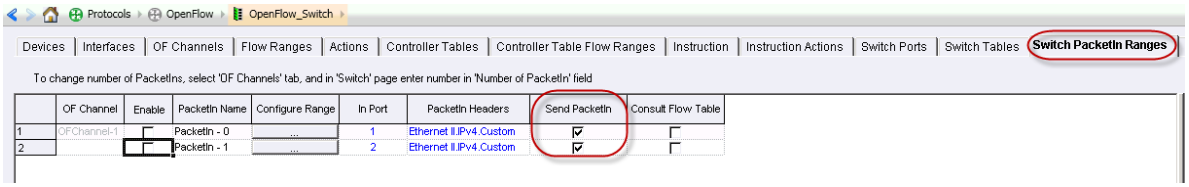
8. Click **OF Channels – Switch** tab and configure following parameters:
   - **Number of Port ranges** as 1
   - **Number of PacketIn ranges** as 2 (This allows you to create different traffic profile)
   - Enable **Calculate PacketIn reply delay** checkbox



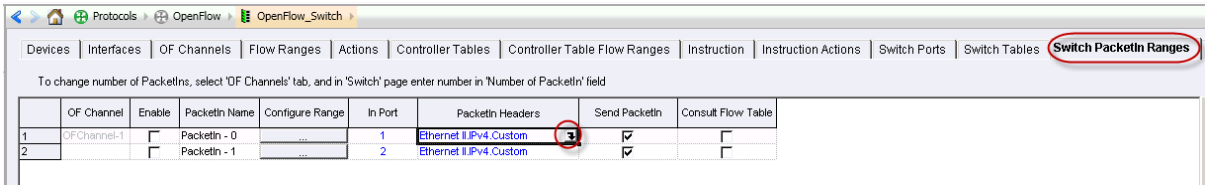
9. In the **Switch ports** tab, configure **Number of Ports** as 10.



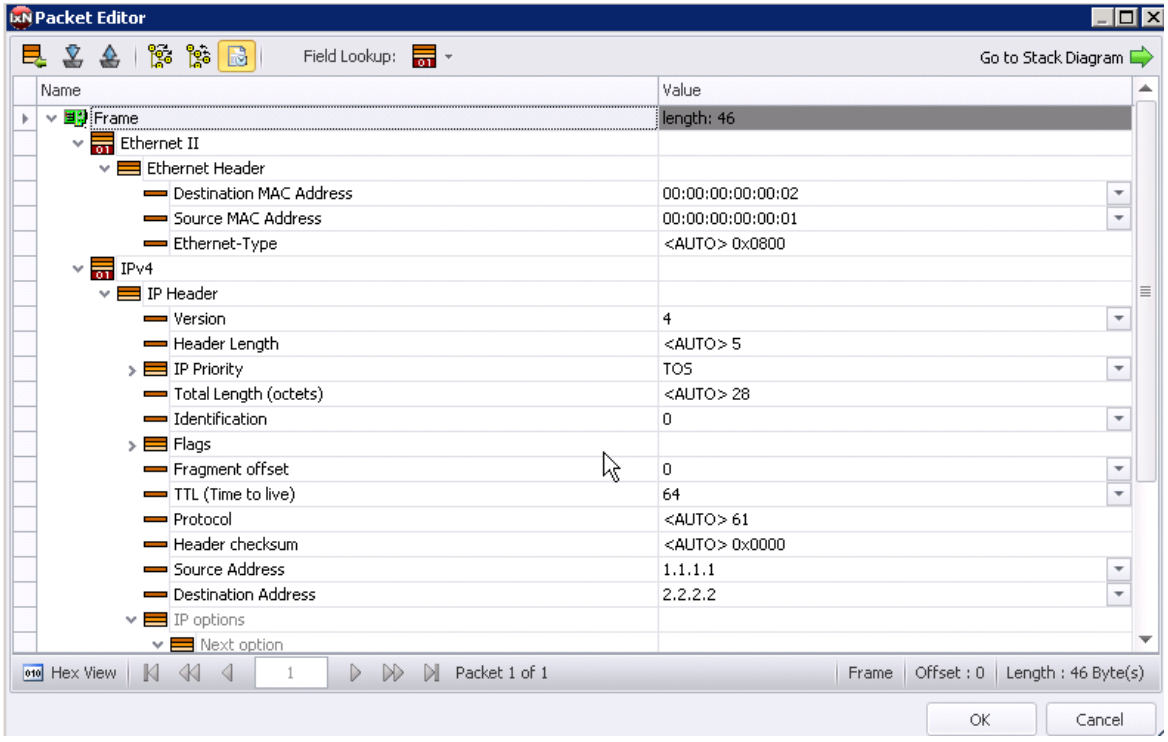
10. Click **Switch PacketIn Ranges** tab. This tab allows user to create various packetIn traffic profiles. Select **Send PacketIn** checkbox and clear **Enable** checkbox.



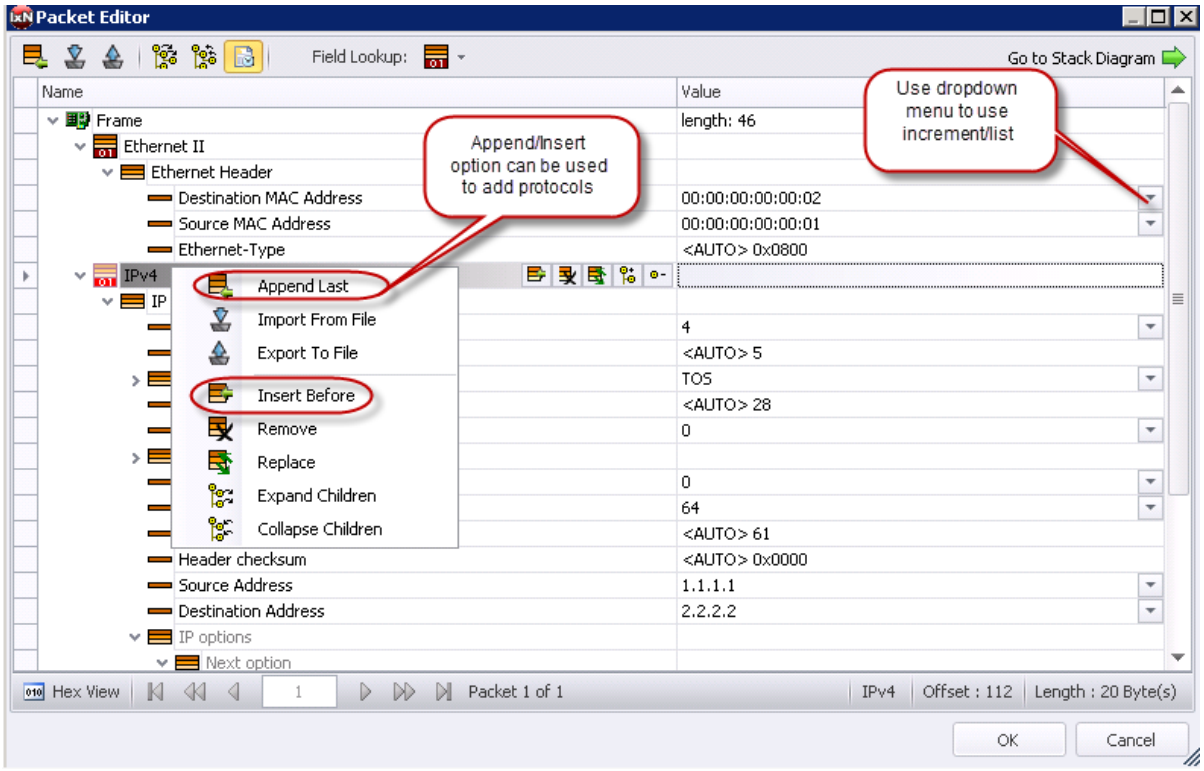
11. In the **PacketIn Headers** column, start Packet Editor by clicking the down arrow (right corner of packetIn headers field) as depicted in the following image.



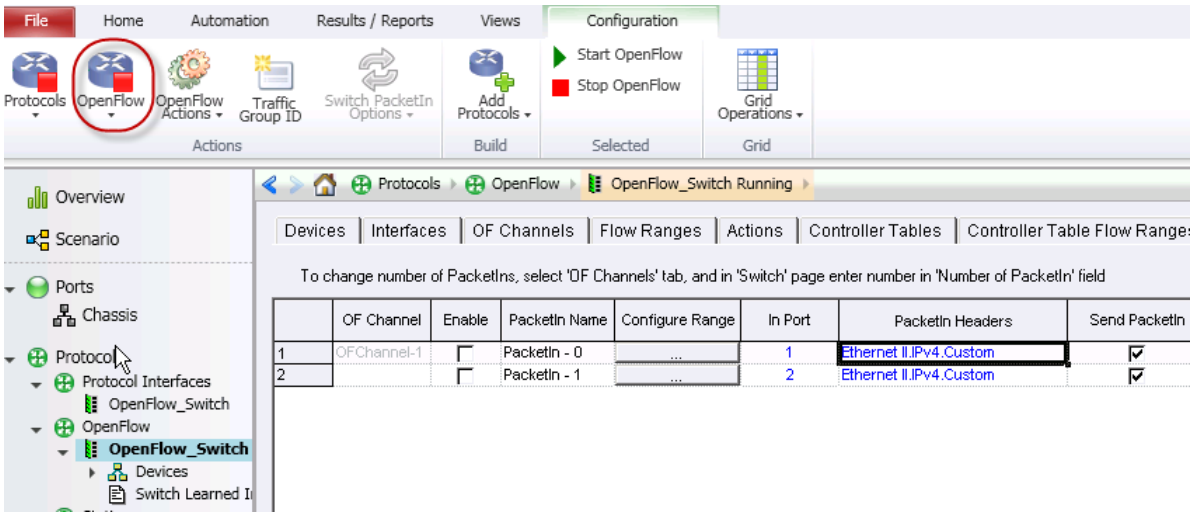
The following image shows sample packet_In packet for IP traffic.



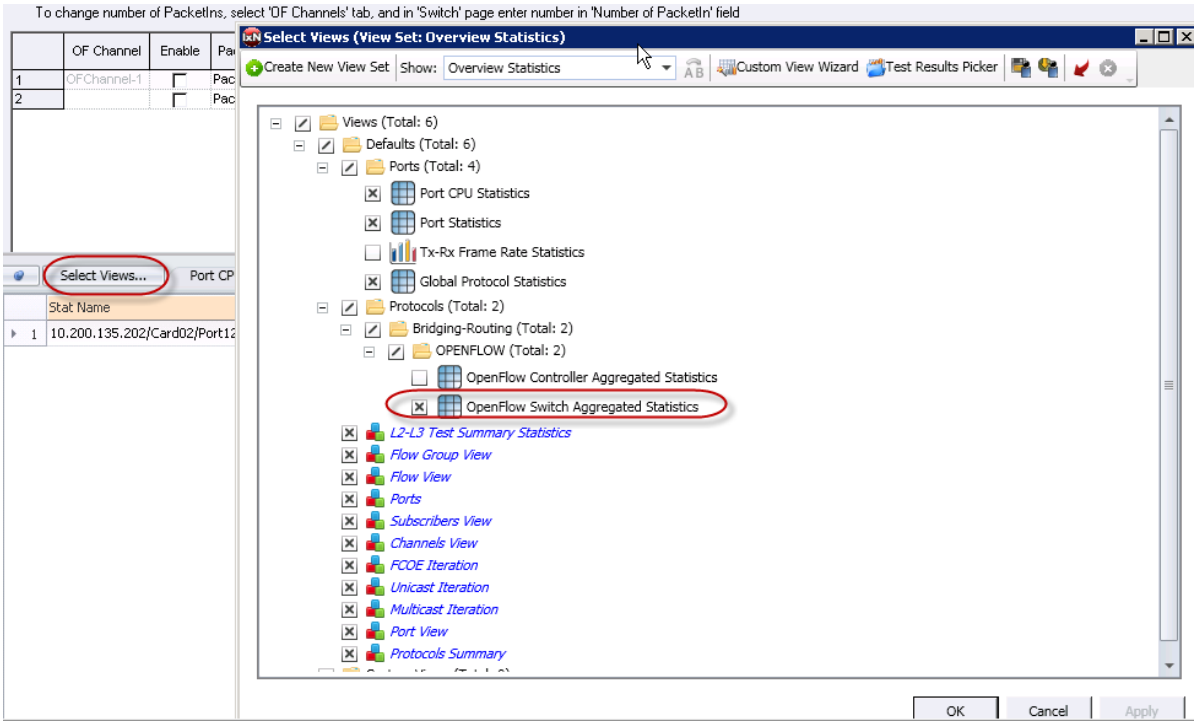
You can use the following options to create different packet types



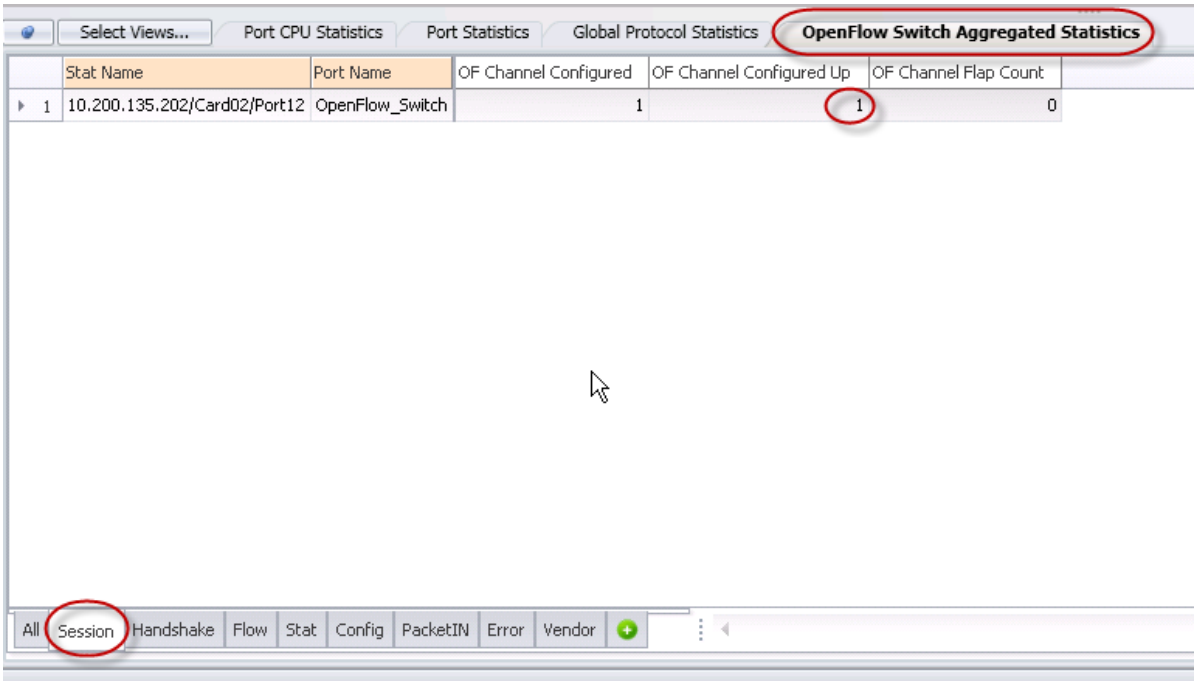
12. Start **OpenFlow** protocol



13. Select the **OpenFlow Switch Aggregated Statistics** checkbox to view the stats.

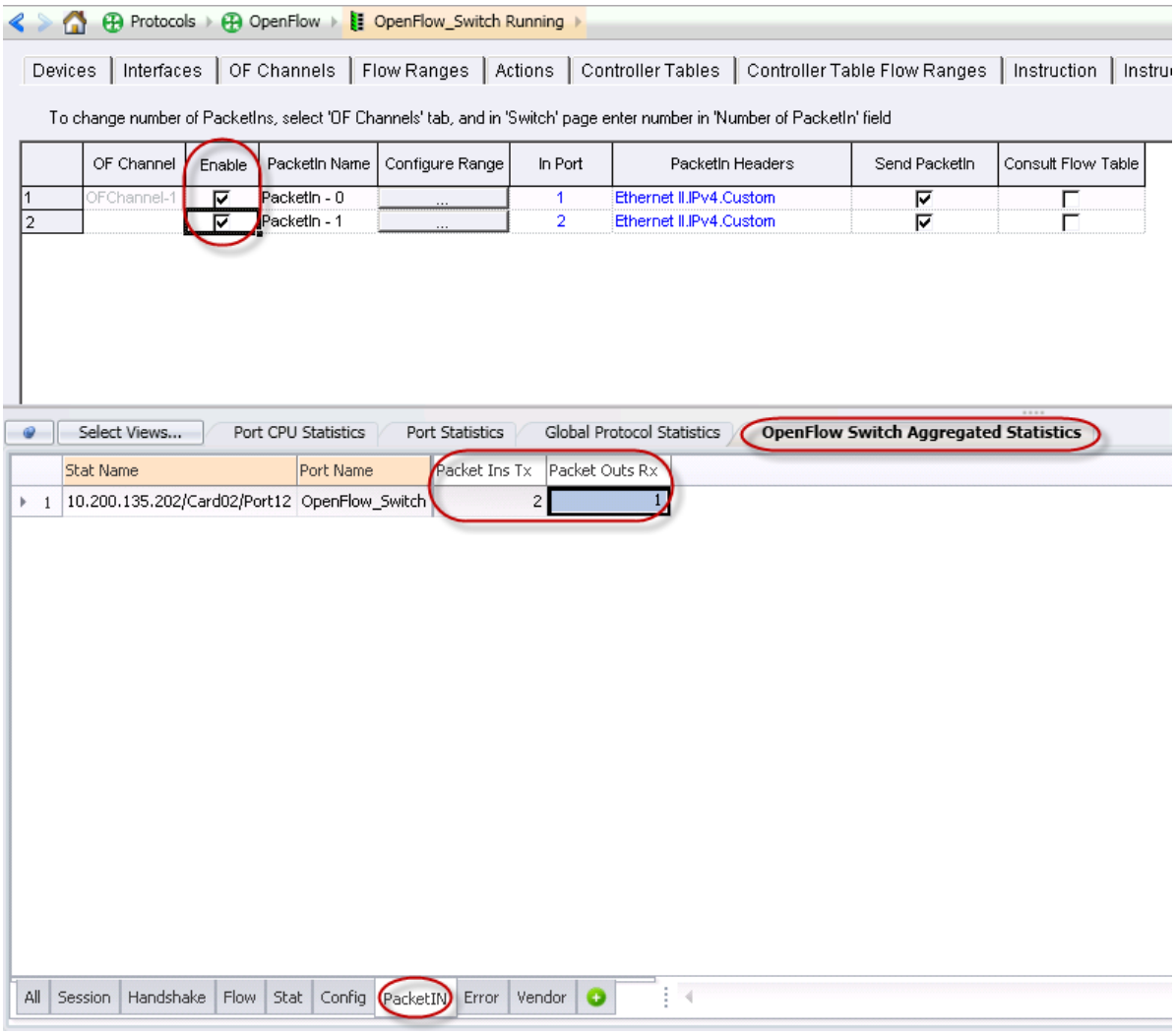


14. Make sure **OF channel** is UP.

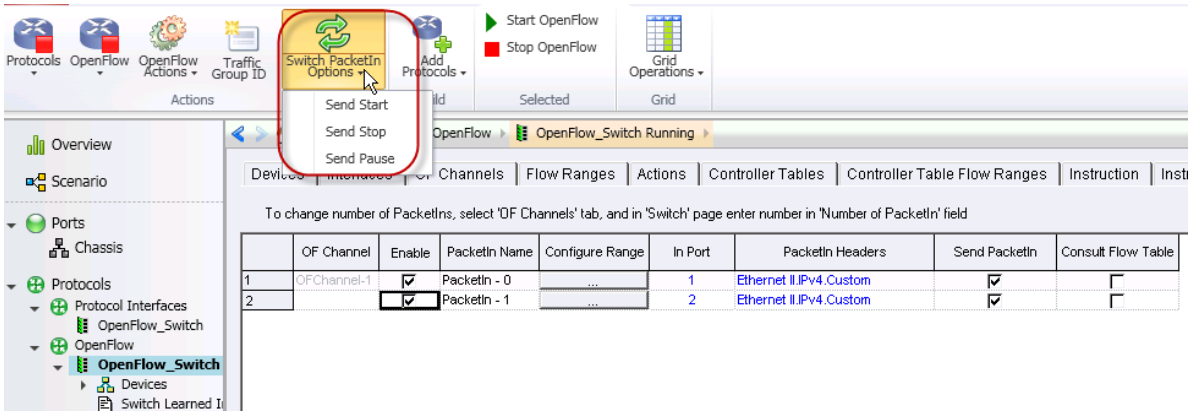


15. Click the **Switch PacketIn Ranges** tab, and select the **Enable** checkbox. This generates packet_in message to the controller and monitor the stats.

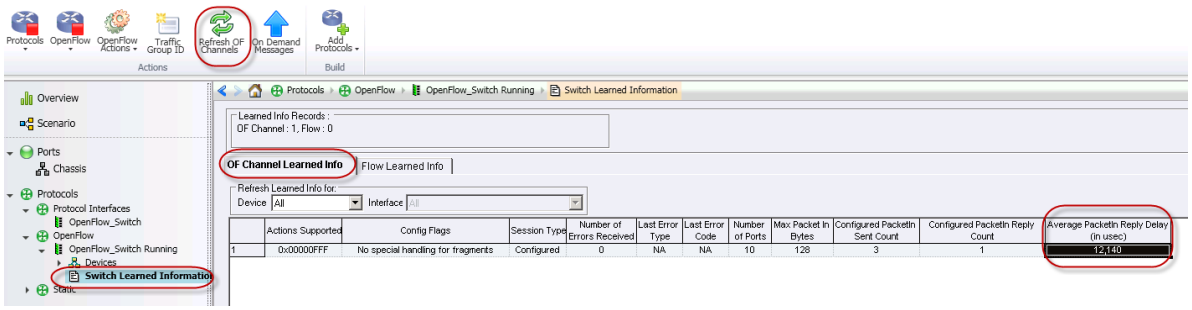
    Note: You can use this enable checkbox while protocol is in Running state. This allows user to generate on-demand packet_in message while protocol is running.



As an alternative, you can also use **Switch PacketIn Options** button in the ribbon to start/stop or pause packet_in message

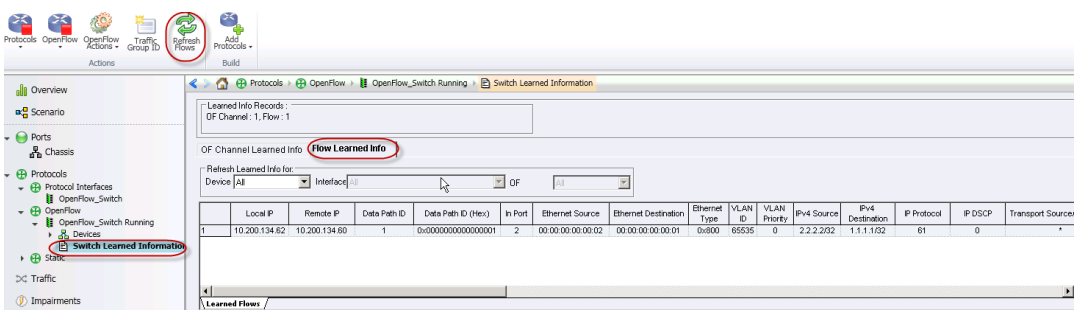


16. From the protocol tree, expand OpenFlow view and click **Switch Learned Information**. Then click **Refresh OF channel** button in the ribbon. This action displays OF channel information and most importantly, it displays **Average PacketIn Reply delay** calculation in microsecond.



Note: This is a cumulative statistics, therefore the calculation is based on total packet_in/packet_out sent/received over the period. If your test case requires to measure controller's response time under certain condition then stop/re-start the OpenFlow protocol to get the response time for that period.

17. To verify whether controller has accurately pushed the flow, click the **Flow Learned Info** tab, and then click on **Refresh Flows** button in the ribbon.



Conclusions

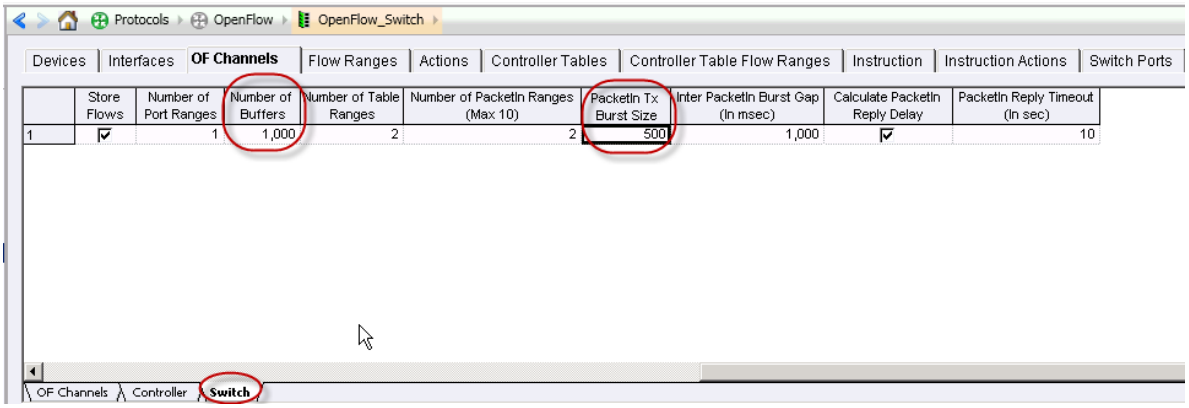
This test case validates:

- How fast the controller is processing incoming packet_ins and sending out packet_out or flow_mod
- Controller's ability to dynamically learn various packet_in types and accurately push the flows to correct switch and the port

Test Variables

Use the following variables to verify the behavior of an OpenFlow controller.

1. Increase the **PacketIn Tx Burst size.** To do that, increasing the **Number of Buffers** count is required.



2. Change **Inter PacketIn Burst Gap** settings to see if it has any effect in controller's responsiveness.
3. Also, try changing **PacketIn Reply Timeout** setting to see it changes the results
4. Create multiple packetIn ranges with different headers and unique packet count.

Test Case: Bandwidth Rate Limiting and QoS validation

Overview

As SDN gradually makes its way into data center networks, the end users' expectation is that it must meet or exceed the benefits offered by traditional networks. So, how can OpenFlow be used to perform rate limiting and QoS traffic engineering?

The OpenFlow v1.3 standard has added Meter/Band functions to support various simple QoS operations such as rate-limiting, QoS remarking, or packet drop. A meter measures the rate of packets assigned to it, and enables control of the packet rate. When installing flows, a controller can attach meters directly to each flow entry, as opposed to queues that are associated to ports.

A meter entry contains the following fields:

- Meter Identifier - A 32-bit unsigned integer uniquely identifying the meter.
- Meter Band - Each meter has one band. The band specifies the rate at which the band applies and the way packets should be processed. If the current rate of packets exceeds the rate of the band, the packets are processed in the way specified by the band.

A meter band contains the following fields:

- Band Type - Defines how packets are processed. Packets that exceed the band rate are dropped or remarked.
- Rate - Defines the lowest rate at which the band can apply.

Objective

This test case helps users to validate meter/band implementation on an OpenFlow-enabled switch. From Ixia's emulated OF Controller we will push high- and low-priority flows with different band types as shown below.

High-Priority flow

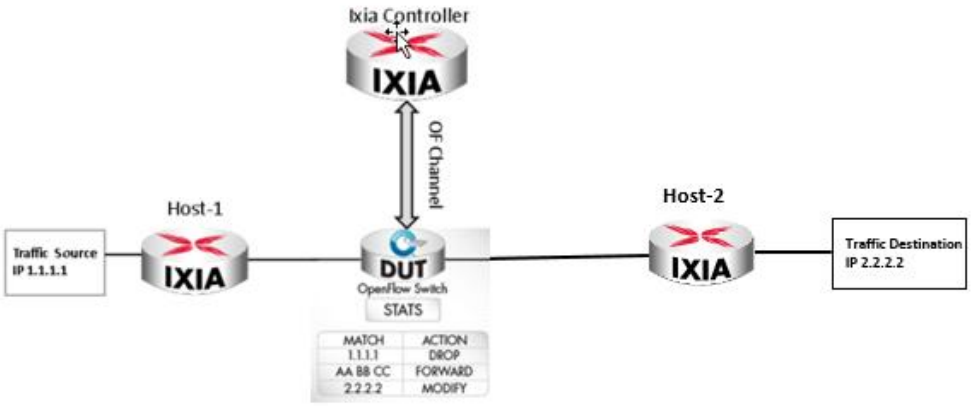
Match = Destination IP address and DSCP = 48
Priority = 1
Instruction = Meter
Band Type = DSCP Remark
Band Rate = 100 Mbit/sec
Apply-Action = Set-field (DSCP value = 0)

Low-Priority flow

Match = Destination IP address and DSCP = 0
Priority = 1
Instruction = Meter
Band Type = Drop
Band Rate = 250 Mbit/sec

The expectation is, when a switch receives data-plane traffic that matches the rate, it should continue to forward traffic without any packet drop or QoS remarking. When the traffic exceeds the configured rate, for high-priority flow, the switch should start remaking the exceeded packets (above the specified rate) and for low-priority flow, it should start dropping the exceeded traffic (not all)

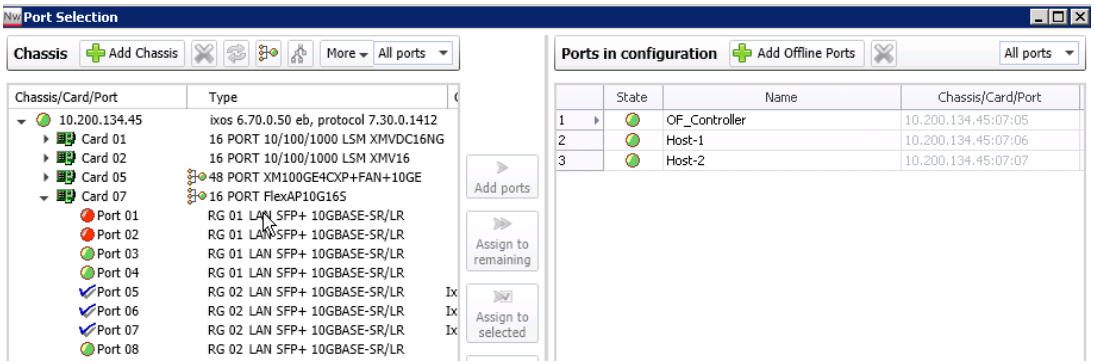
Setup



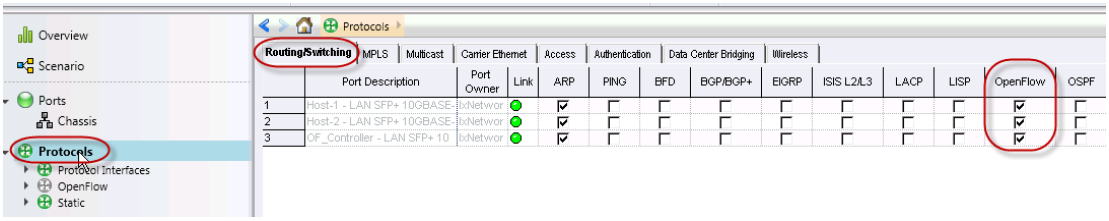
Step-by-Step Instructions

The following steps describe how to apply meter/band to each flow entry and validate QoS implementation of an OpenFlow switch

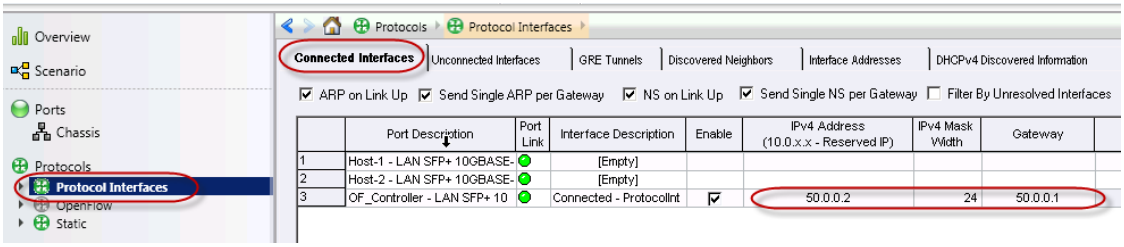
1. **Reserve 3 Ixia ports** (1 for OF Controller and 2 for data-plane traffic)



2. In the **Protocols** Window, select the **OpenFlow** checkbox to enable OpenFlow

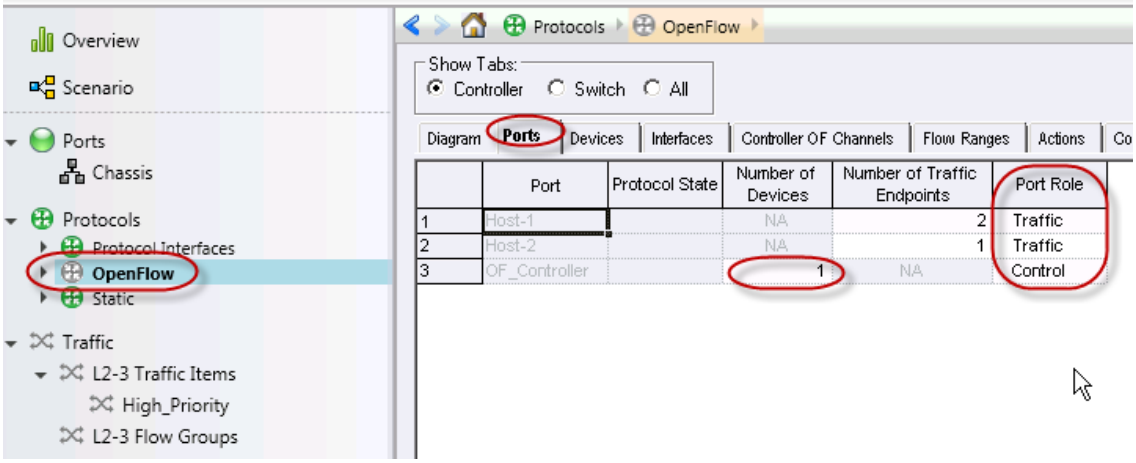


3. Click the **Connected Interface** tab to configure the emulated OpenFlow switch, IP address, and **Gateway** address in the **Protocol Interfaces** window. For OF Channel, ensure that ARP is resolved.

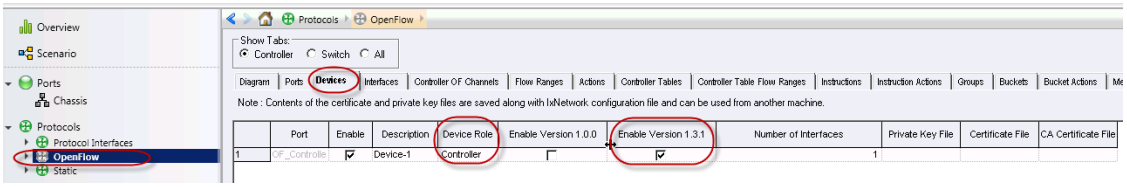


4. Define the **Number of Devices** and the port role as **control** by selecting the role from the **Port Role** list on the **Ports** tab on the **OpenFlow** window.

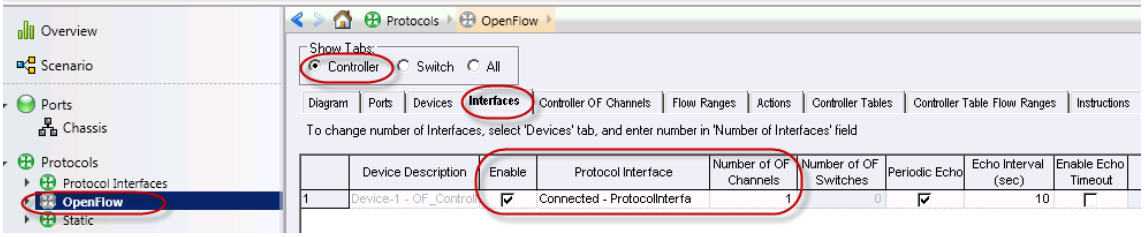
   Note: The Number of Devices option allows creating multiple OpenFlow Switch emulations on a single physical Ixia port. Make sure a unique protocol interface is created and assigned to each emulated switch.



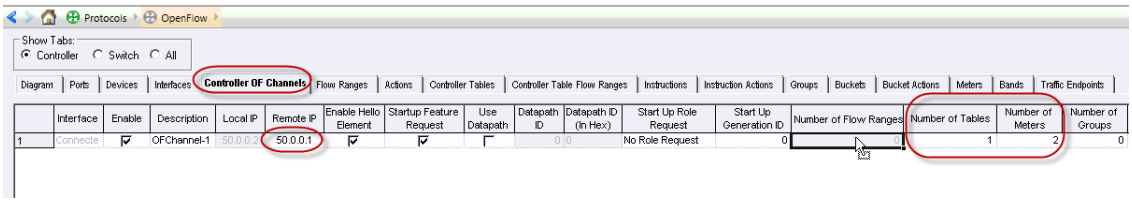
5. Click the **Devices** tab and configure the following parameters:
   - **Device Role** as Controller
   - **Enable version** v1.3
   - **Number of Interfaces** as 1
   - If a secured TLS OF channel connection is desired then specify key file path **(Optional)**



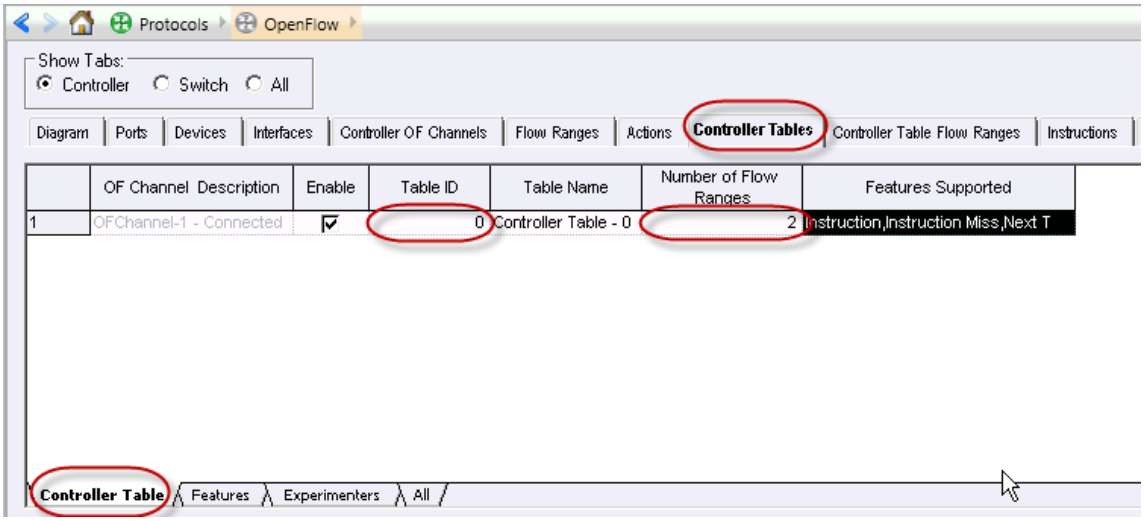
6. Click the **Interface** tab of the **OpenFlow** window and assign the **Protocol Interfaces** that you created in the **Protocol Interface** window. This interface is used for the control-plane (OF Channel). Configure **Number of Channels** as *1*



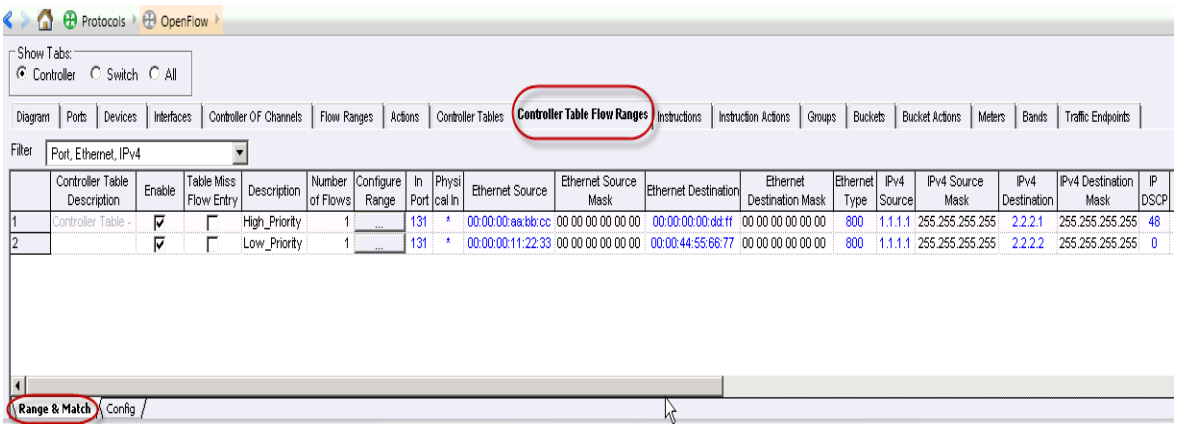
7. In the **OF Channels** tab specify the IP address of the OpenFlow switch in the **Remote IP** field. Also, configure **Number of Tables** as 1 and **Number of Meters** as 2



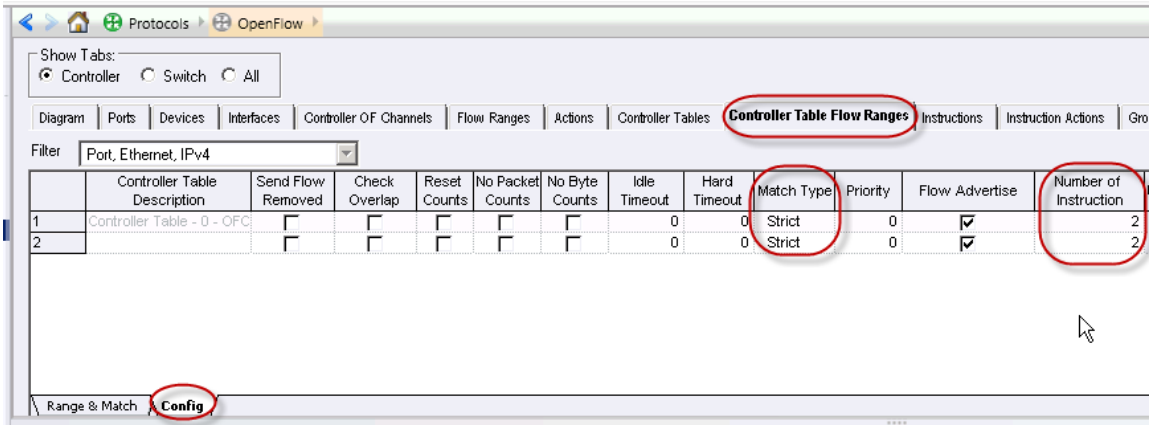
8. Go to the **Controller Tables** tab and configure **Table ID** 0 (default) and **Number of Flow Ranges** to 2



9. In **Controller Table Flow Ranges** tab configure 2 flows with **Src/Dst MAC and IP address, Ether type and DSCP value**.

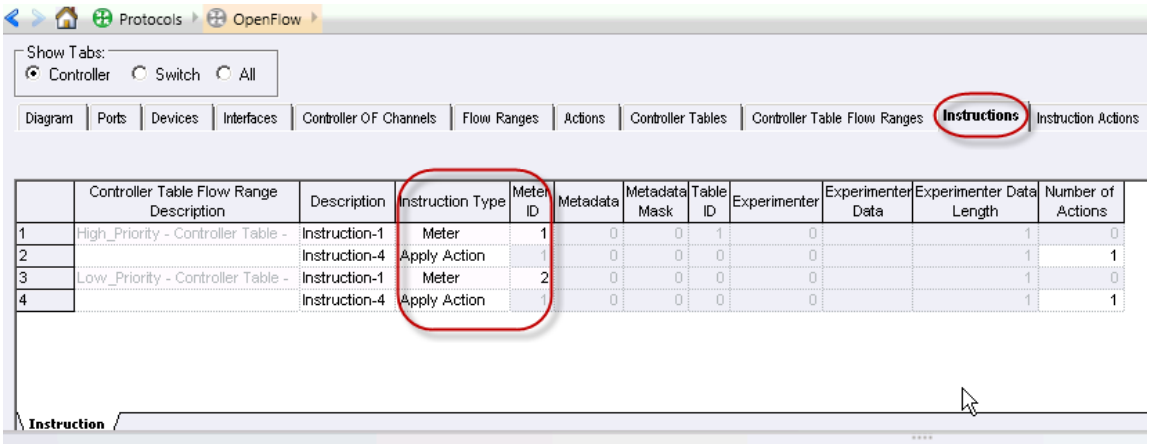


10. On **Controller Tag Flow Ranges > Config** tab, change **Match type** to Strict and **Number of Instruction** to 2

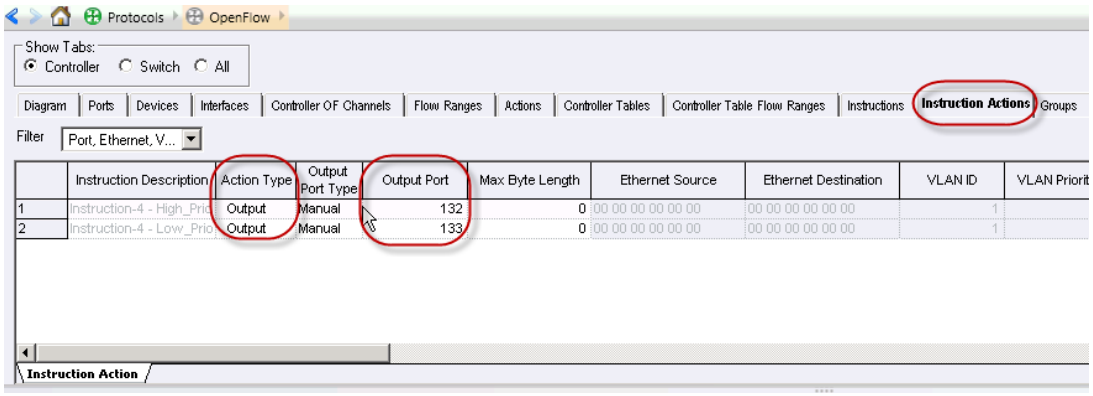


11. Go to **Instructions** tab, change **Instruction Type** as Meter and configure **Meter ID**

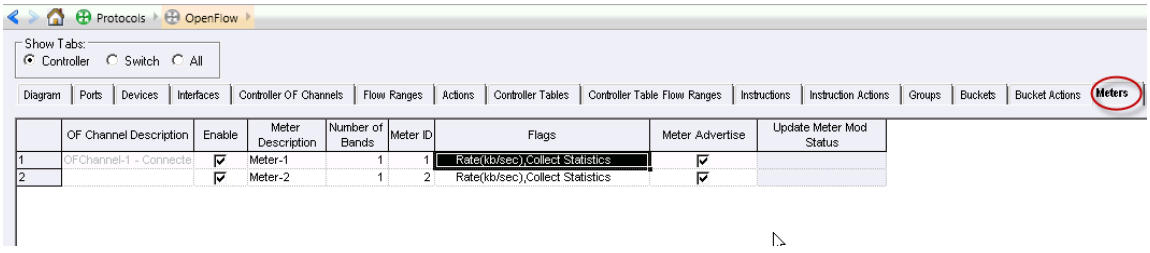
Note: The Meter ID must be the same as per Meters tab



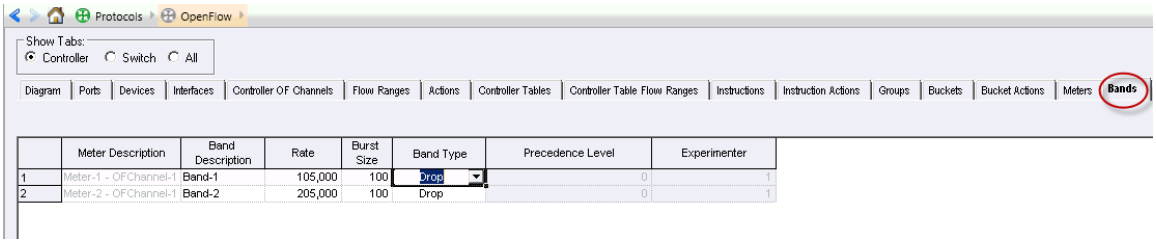
12. On **Instruction Actions** tab, change the **Action Type** as Output and correctly specify **Output port**



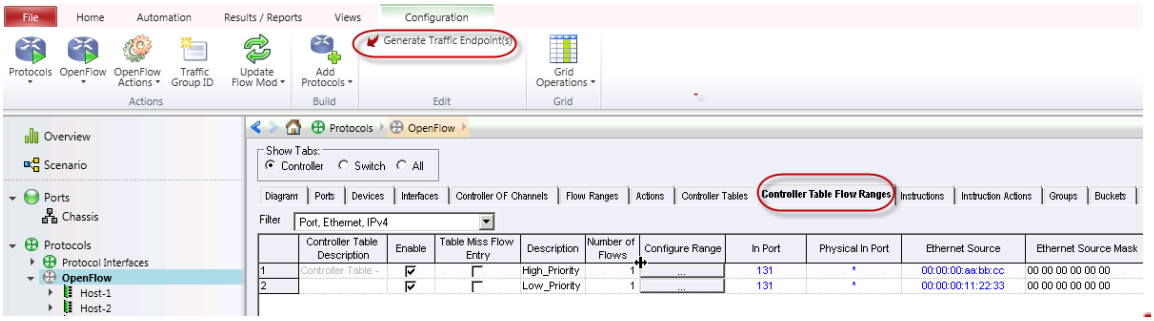
13. On **Meters** tab, configure **Meter ID**, enable **Rate (Kb/Sec) and Collect Statistics** flags



14. Go to **Bands** tab, configure desired **Rate** and set the **Band Type**

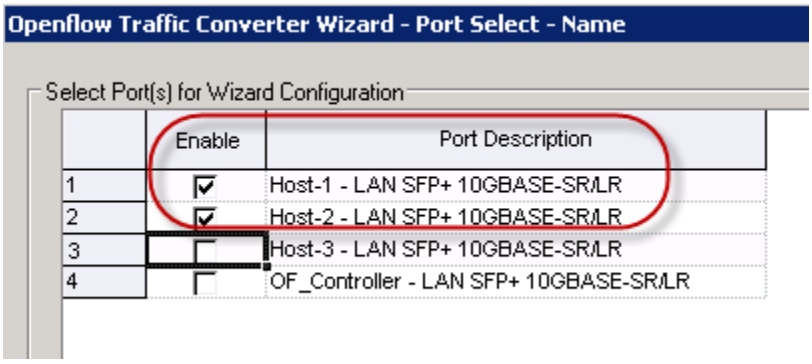


15. From **Controller Table Flow Ranges** tab launch **Generate Traffic endpoints** wizard. This step is required to create traffic endpoints with matching source/destination flow entry

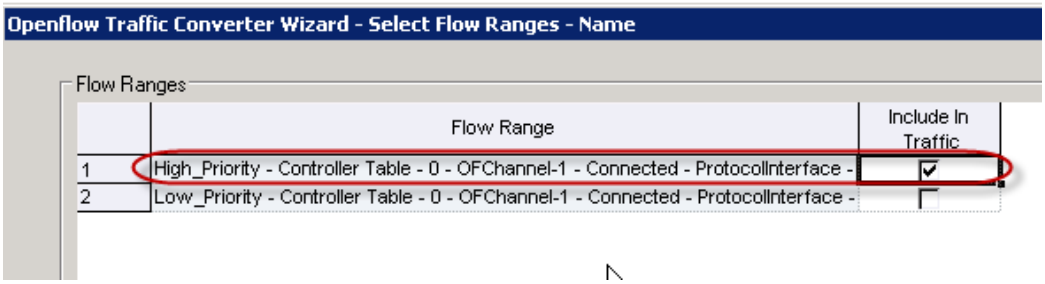


16. **Generate Traffic endpoints** wizard steps:

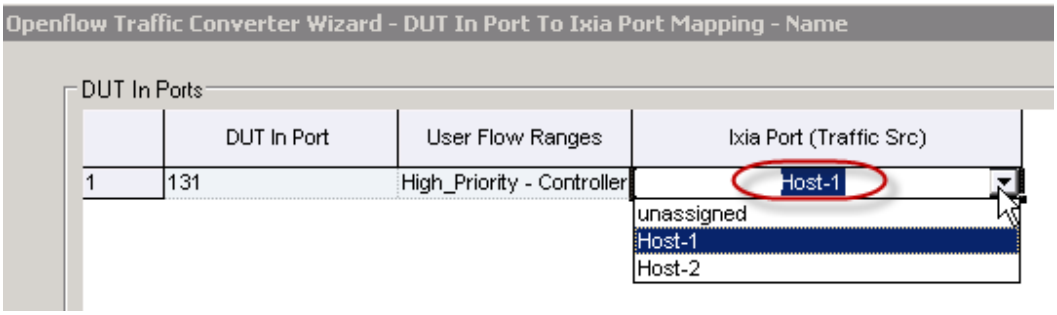
    a. On page#1, select **Source/Destination port** and hit **Next**



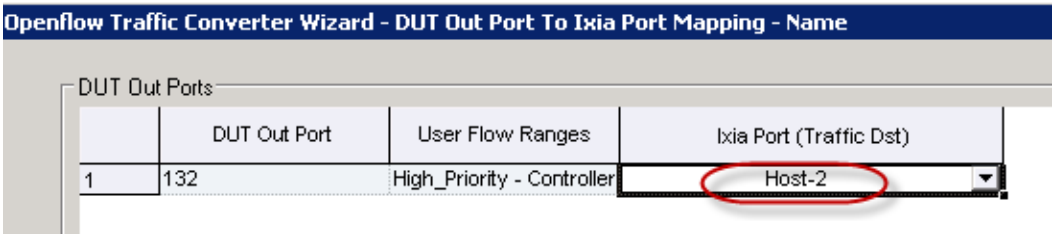
b. On page#2, select the **flow range (**in this case **High Priority** flow range**)** that we want to create **traffic endpoints** and hit **Next**



c. On page#3, select the **Traffic Source** port and hit **Next**

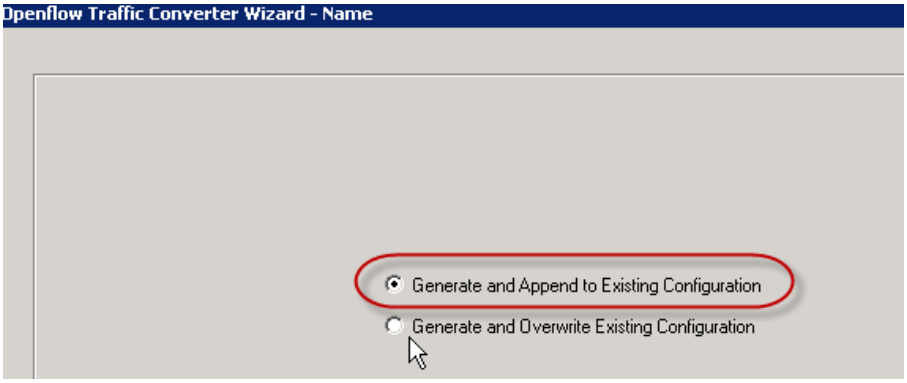


d. On page#4, select **Traffic Destination** port and hit **Next**



e. On page#5 and 6 verify **source and destination field** information to make sure it is correct

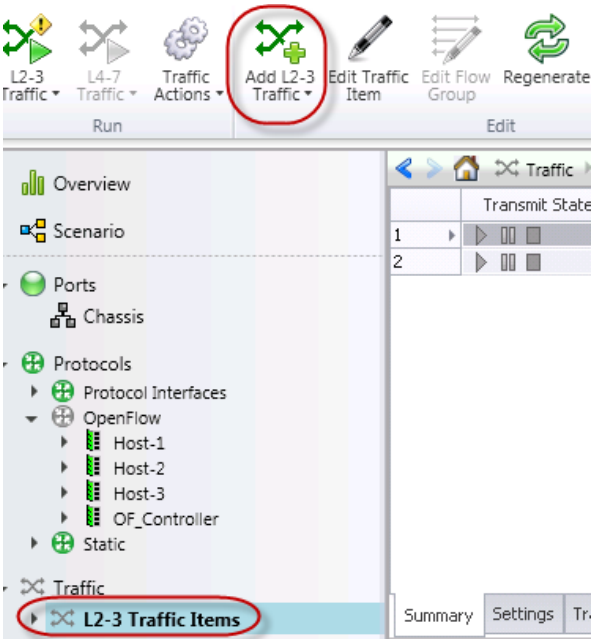
f. On page#7, select right option to apply configuration. The Generate and Append option will add a new traffic endpoint (i.e., it will keep existing endpoints on that port). The 2nd option, Generate and Overwrite, will erase existing endpoints and create new endpoints. So, carefully select this option.



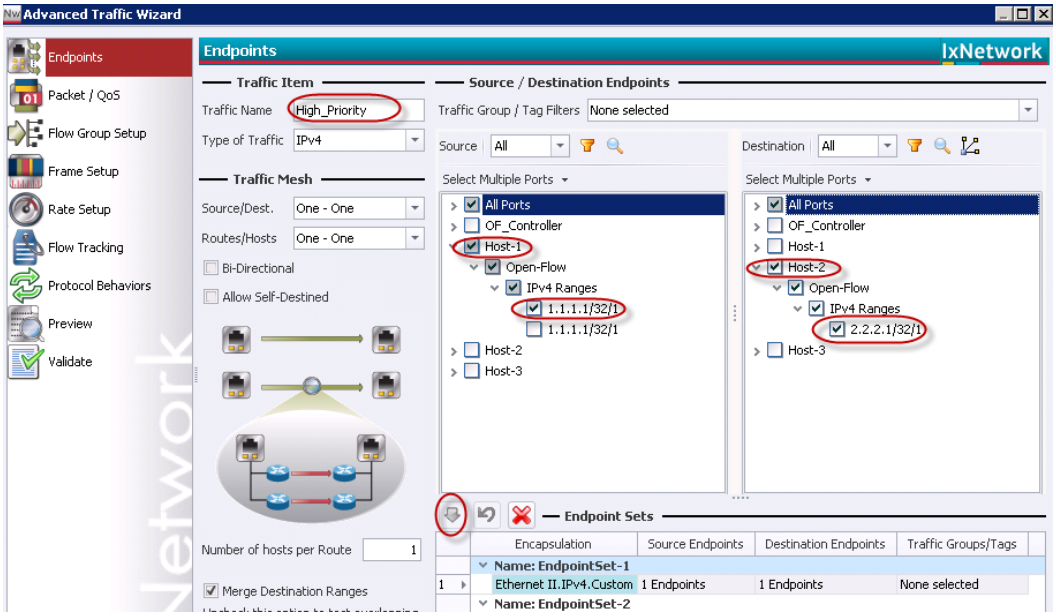
g. Repeat steps **a to f** for another flow range(in this case **Low Priority flow** range)

17. Once the traffic endpoints are created, Use Traffic Wizard to create a traffic stream for High- and Low-Priority flows by following these steps:

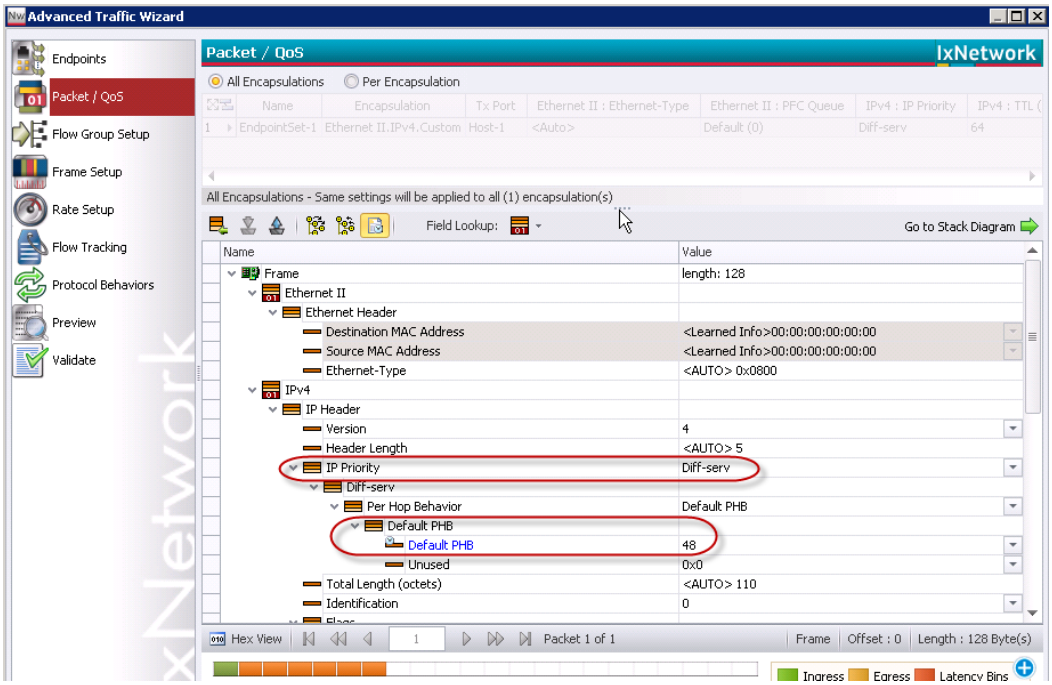
a. Launch traffic wizard



b. Select source and destination endpoint and hit next



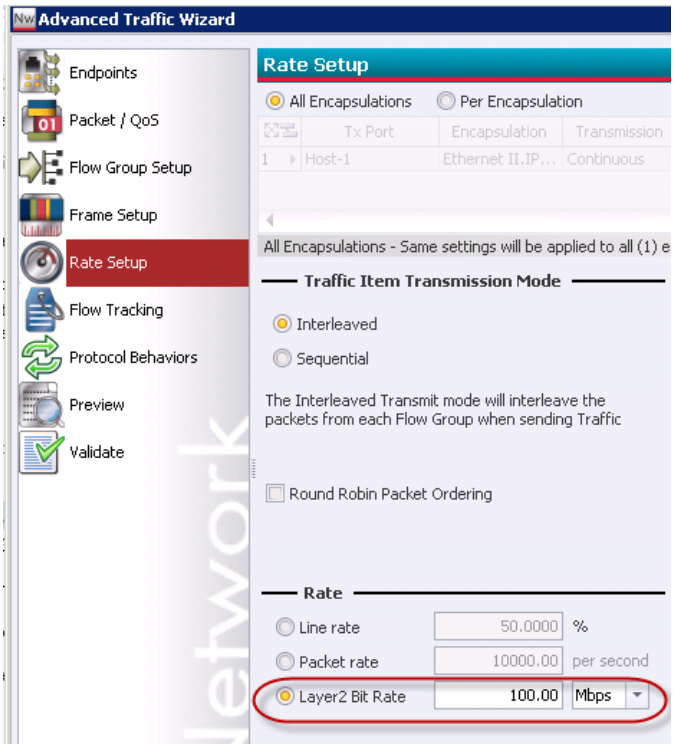
c. On page#2, change IP priority to **Diff-serv** and set **DSCP** value



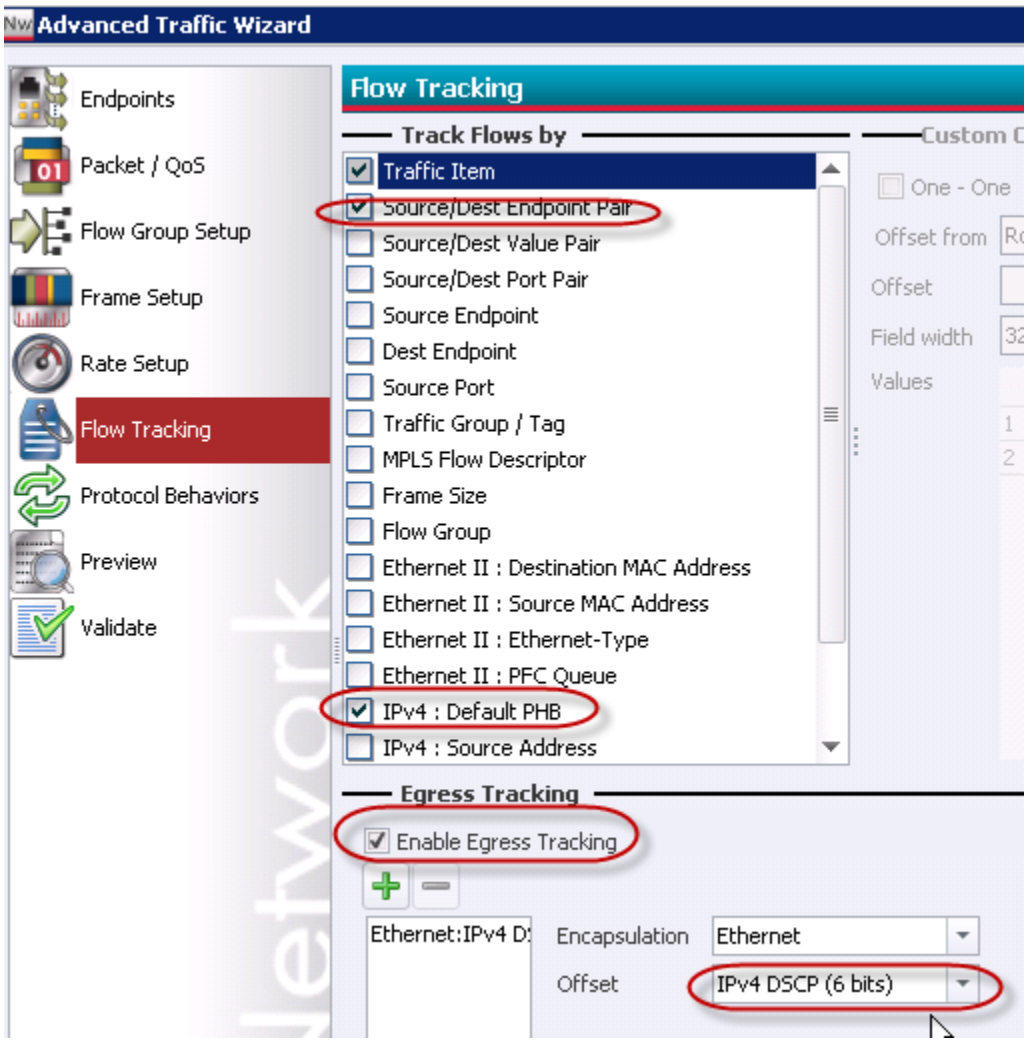
d. Leave **Flow Group setup** page as it is

e. On **Frame Setup** page, specify desired Frame Size

f. On **Rate Setup** page, configure L2 packet rate to matching band rate (in this case for High Priority flow, we are setting 100Mbps)



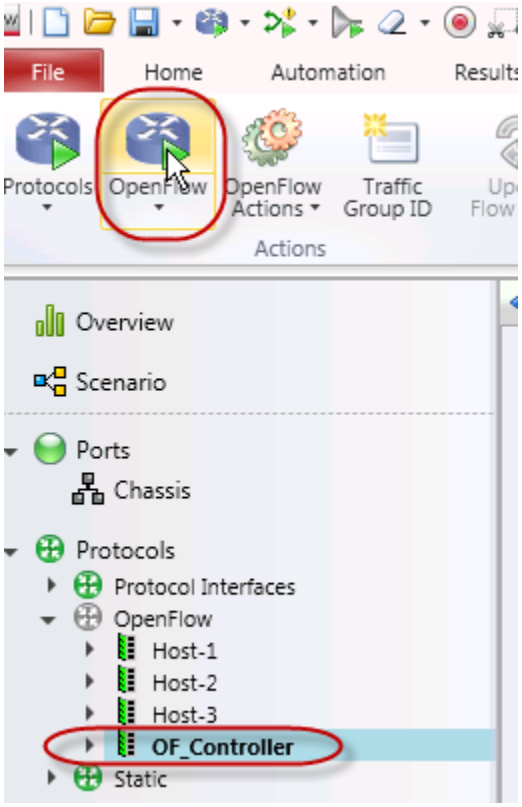
g. On **Flow Tracking** page, enable Ingress tracking for **source/destination pair, IPv4 PHB**. Also, enable **egress tracking** for DSCP



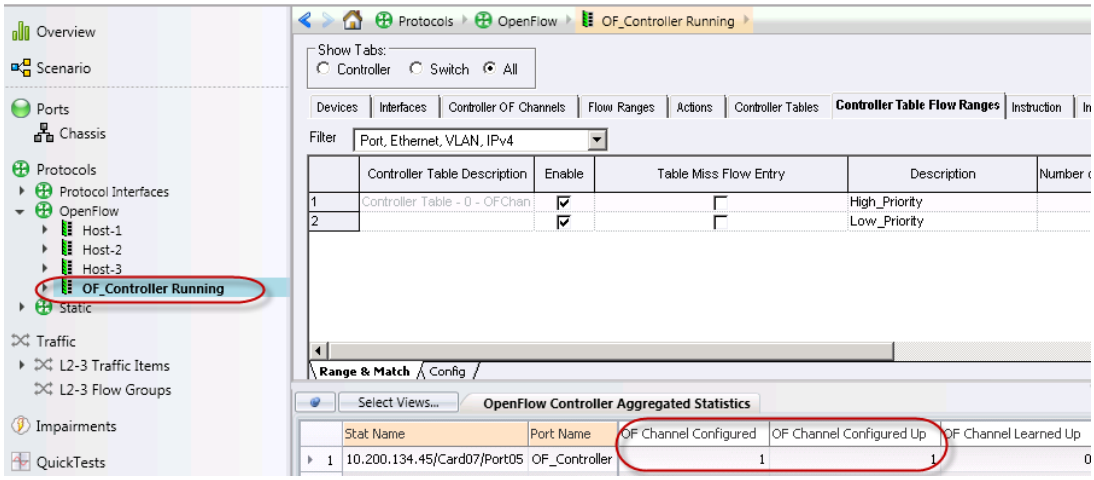
h. Leave all other page defaults and Finish the wizard

i. Build traffic stream for **Low Priority** flow by repeating steps **a to h**. For Low Priority traffic, use **Diff-Serv value 0** and **L2 Bit Rate 200 Mbps**

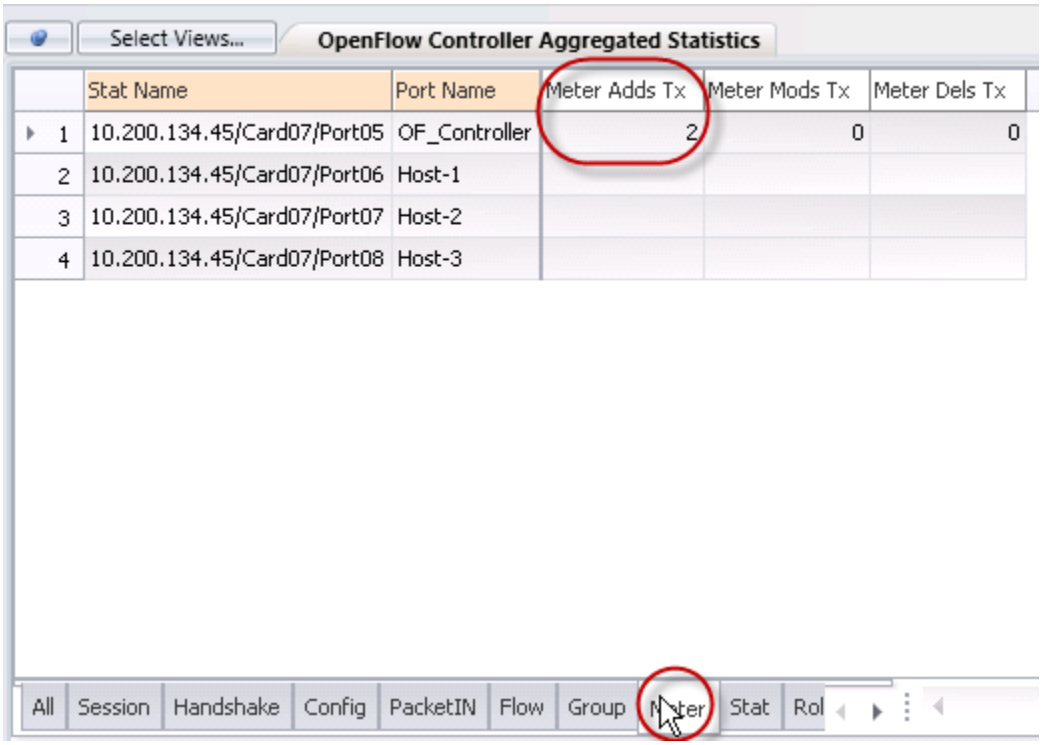
18. **Start OpenFlow** protocol



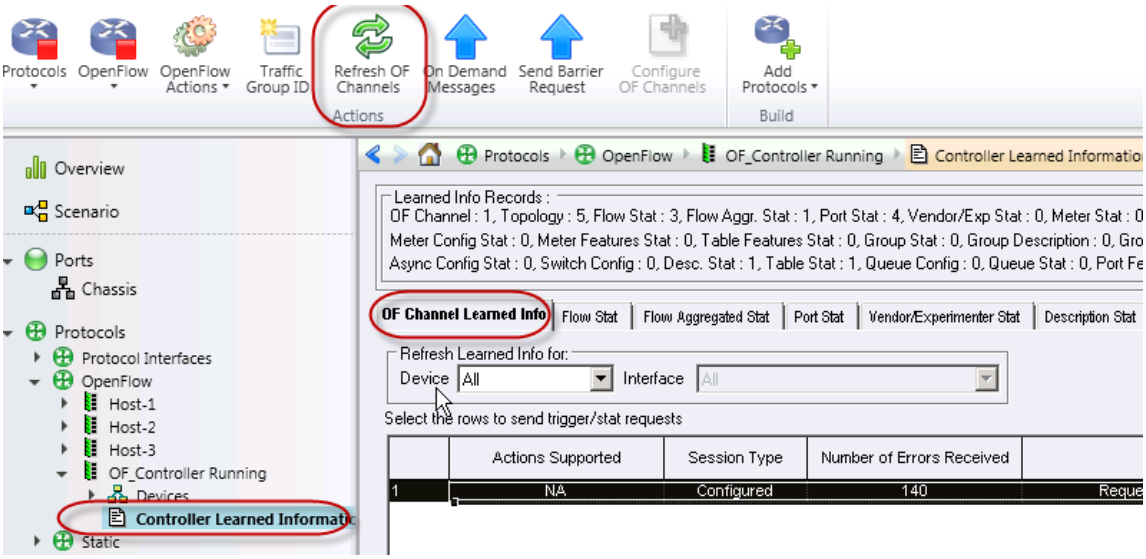
19. Make sure **OF channel** comes up



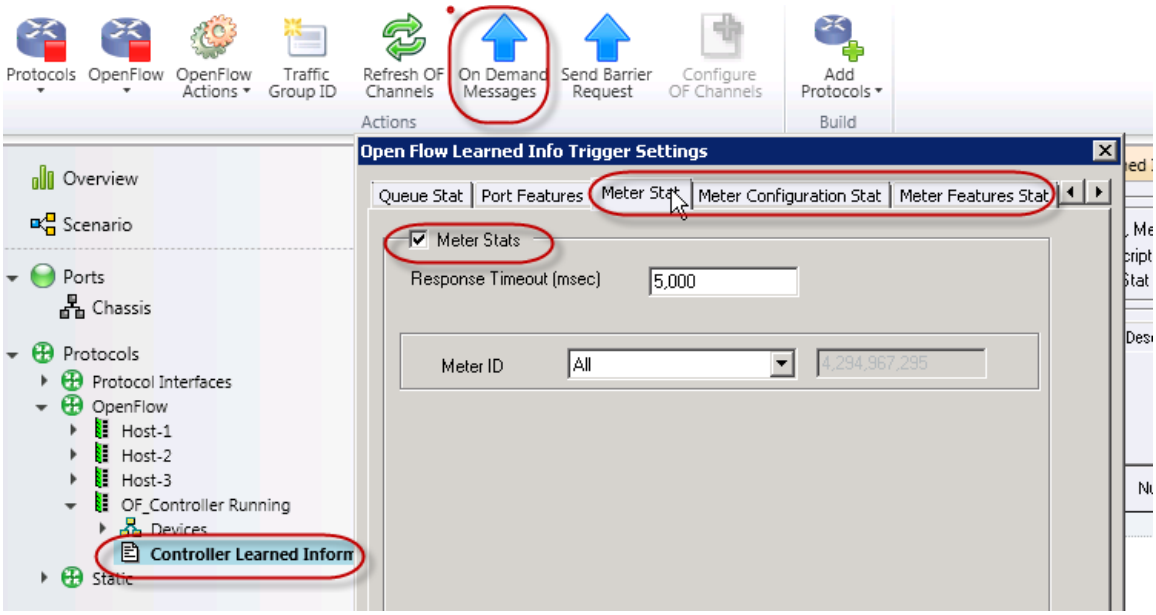
20. Also, check **Meter stats** to see if OF Controller added Meter along with flow entry



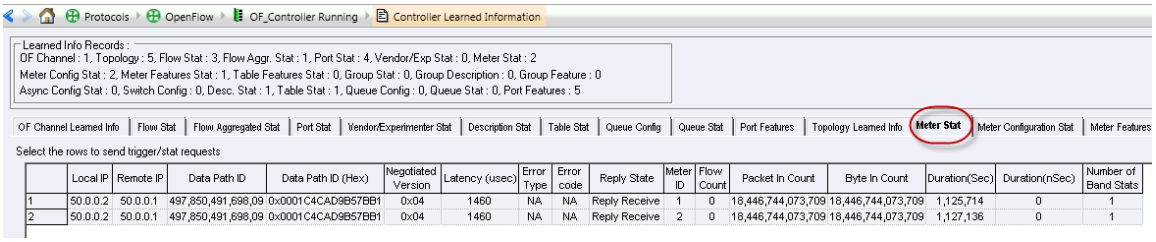
21. From **Controller Learned information, Refresh OF Channel** to get status

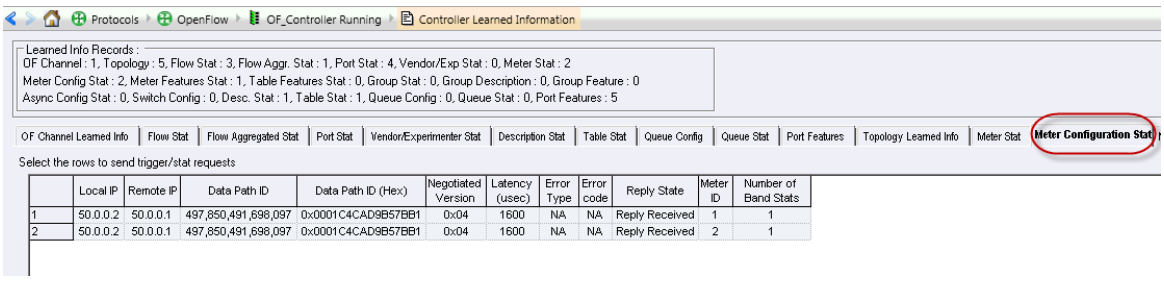


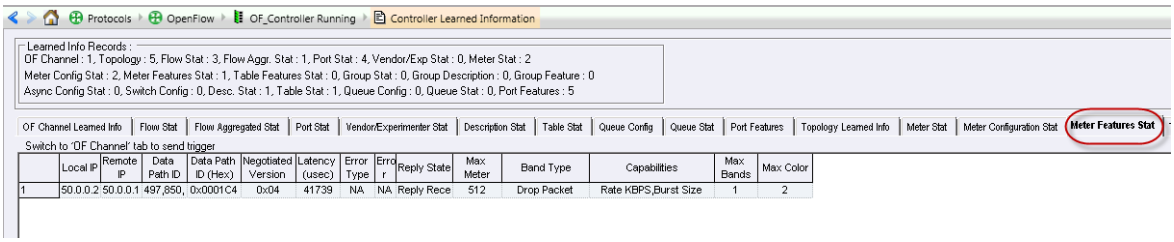
22. Trigger on-demand **Meter stats**, **Meter Configuration stat,** and **Meter Features stat**



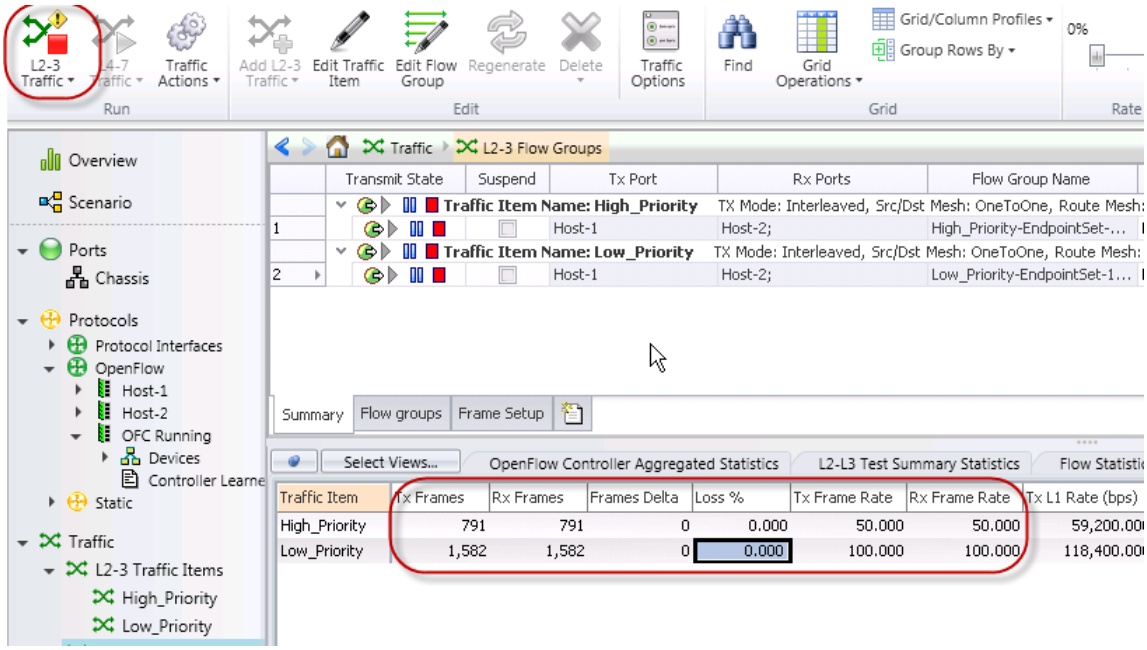
23. Analyze **Meter Stat, Meter Configuration Stat, and Meter Features Stat** response from the switch



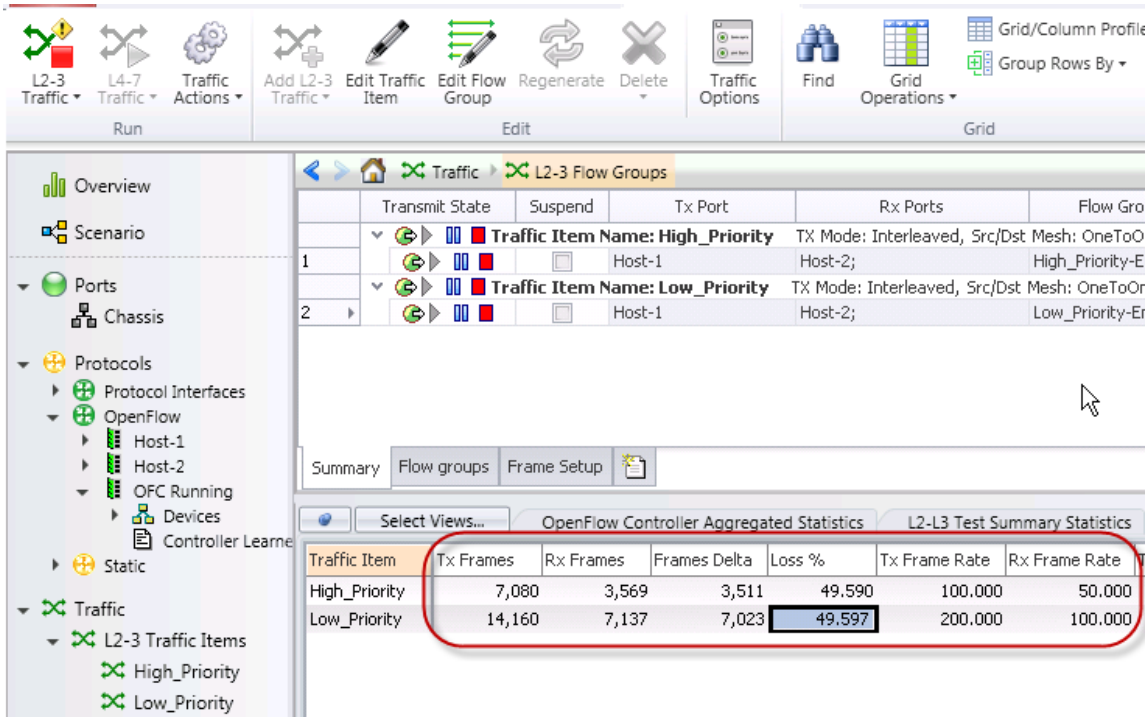




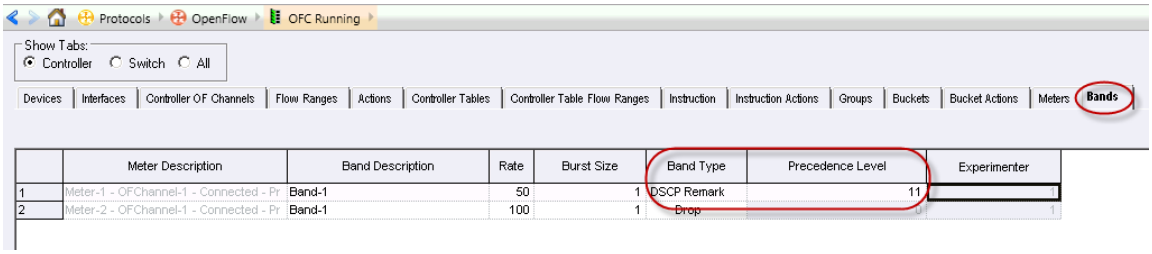
24. **Start data-plane traffic**, since traffic rate is below (or equal) to configured rate, no traffic loss should be observed



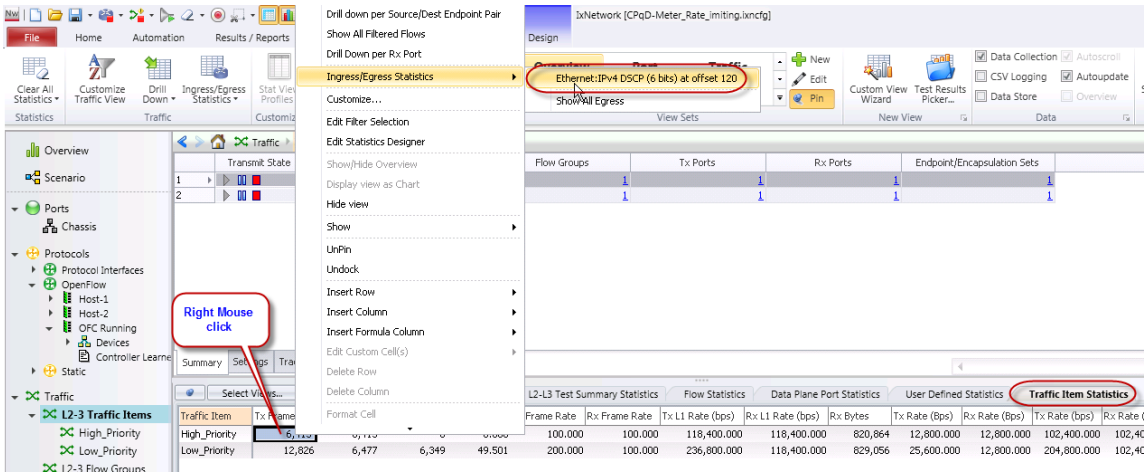
25. Now, **double the traffic rate** on both streams. Since the rate is exceeding configured Meter rate, switch should drop 50% traffic



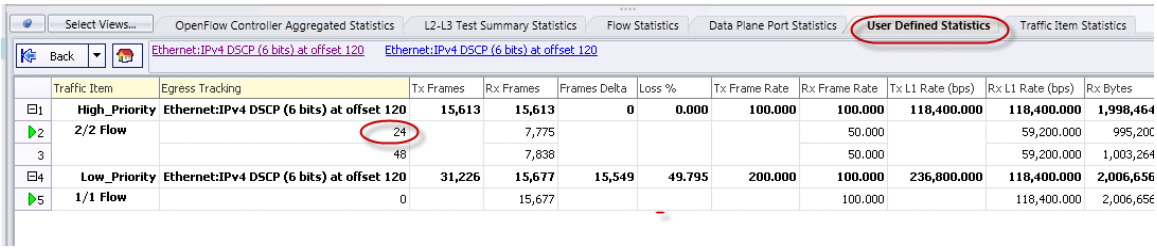
26. To further test Meter function, try to change band type as DSCP remark



27. Start data-plane traffic with **double rate** than configured Meter rate. Observe **Ingress/Egress** statistics.



The switch should start performing **DSCP remark** on exceeded packets as shown in below snapshot



Conclusion

This test case validates meter functionality. When the traffic rate for the associated flow entry stays below the configured rate, the switch should not apply meter/band and no packet loss should be observed. When the traffic goes above the specified rate, the switch should apply meter and should either drop the exceeded traffic or perform DSCP remark for exceeded traffic if the band type is set to DSCP remark.

Contact Ixia

Corporate Headquarters
Ixia Worldwide Headquarters
26601 W. Agoura Rd.
Calabasas, CA 91302
USA
+1 877 FOR IXIA (877 367 4942)
+1 818 871 1800 (International)
(FAX) +1 818 871 1805
sales@ixiacom.com

Web site: www.ixiacom.com
General: info@ixiacom.com
Investor Relations: ir@ixiacom.com
Training: training@ixiacom.com
Support: support@ixiacom.com
+1 877 367 4942
+1 818 871 1800 Option 1 (outside USA)
online support form:
http://www.ixiacom.com/support/inquiry/

EMEA
Ixia Technologies Europe Limited
Clarion House, Norreys Drive
Maiden Head SL6 4FL
United Kingdom
+44 1628 408750
FAX +44 1628 639916
VAT No. GB502006125
salesemea@ixiacom.com

Renewals: renewals-emea@ixiacom.com
Support: support-emea@ixiacom.com
+44 1628 408750
online support form:
http://www.ixiacom.com/support/inquiry/?location=emea

Ixia Asia Pacific Headquarters
21 Serangoon North Avenue 5
#04-01
Singapore 5584864
+65.6332.0125
FAX +65.6332.0127
Support-Field-Asia-Pacific@ixiacom.com

Support: Support-Field-Asia-Pacific@ixiacom.com
+1 818 871 1800 (Option 1)
online support form:
http://www.ixiacom.com/support/inquiry/