ONUG Network Traffic Monitoring/Visibility Working Group

ONUG/Ixia Test Plan for Top 10 Requirements

August 2015

Table of Contents

Introduction 4

Reference architecture 4

Tests 5

Requirement 1: Commodity hardware based upon merchant silicon with either an open or proprietary Switch OS 6

Validation conditions (Pass) 6

Requirement 2: Granular filtering based on 5-tuple and/or even more, including application signatures, and Quality of Service (QoS) marking capability. 6

Test objective 6

Test architecture 6

Prerequisites 6

Simulated traffic and procedure 6

Expected results 7

Validation conditions (Pass) 7

Requirement 3: Capability to work with both underlay and overlay protocols, providing independent filtering on either, and/or correlate both traffic. 7

Test objective 7

Test architecture 7

Prerequisites 8

Simulated traffic and procedure 8

Expected results 9

Validation conditions (Pass) 9

Requirement 4: Process data without impact to production flow/processing flow (CPU/Memory/Bandwidth) 9

Test objective 9

Test architecture 9

Prerequisites 9

Simulated traffic and procedure 9

Expected results 9

Validation conditions (Pass) 9

Requirement 5: Horizontal scalability with the capability of resource management feedback. 9

Test objective 10

Test architecture 10

Prerequisites 10

Simulated traffic and procedure 10

Expected results 10

Validation conditions (Pass) 10

Requirement 6: Must be able to locally process data and create traffic signaling/alerting, while executing defined traffic based actions. 10

Test objective 10

Test architecture 10

Prerequisites 10

Simulated traffic and procedure 10

Expected results 11

Validation conditions (Pass) 11

Requirement 7: Interoperability between vendors: integration and output that will support data collection integration. Open API for access and management (in/out). 11

Test objective 11

Test architecture 11

Prerequisites 11

Simulated traffic and procedure 12

Expected results 12

Validation conditions (Pass) 12

Requirement 8: Capability of Packet De-duplication/Packet Slicing/Data Masking and Application Recognition, including Packet Organization. 12

Test objective 12

Prerequisites 13

Simulated traffic and procedure 13

Expected results 13

Validation conditions (Pass) 14

Requirement 9: The ability to be Security and Compliance-aware. 14

Test objective 14

Test architecture 14

Prerequisites 14

Simulated traffic and procedure 14

Expected results 15

Validation conditions (Pass) 15

Requirement 10: Multilayer visibility between underlay and overlay protocol use for management/Service-Level Agreement (SLA), monitoring/alerting, troubleshooting and reporting capabilities. 15

Test objective 15

Test architecture 15

Prerequisites 16

Simulated traffic and procedure 16

Expected results 17

Validation conditions (Pass) 17

# Introduction

This document outlines a series of test cases to demonstrate support for the top 10 requirements from the ONUG Network Monitoring/Visibility working group white paper. The goal of the tests is to demonstrate support for each of the requirements. There is one test case per requirement. There is no negative testing apart from when the requirement itself requests it. There are no high-performance test cases either, since validating functionality is the main objective.

Some test cases are labeled “N/A.” This is because it’s not possible to demonstrate the requirement using test tools and/or traffic generators. These requirements are not addressed in this test plan.

Each test case contains the following sections:

* Objective: to state the purpose of the test
* Test architecture: outlines the System Under Test (SUT) and the test tools to be used, and how they are networked.
* Prerequisites: a high-level description of the state of the SUT and the test tools before the test starts
* Simulated traffic and procedure: a description of the traffic to be used from the test tools and the procedure to be followed for the test
* Expected results: a high-level description of the behavior should the test succeed
* Validation conditions: the criteria for a pass or fail for the test case

The test tool to be used for this test plan is Ixia’s IxNetwork (virtual edition), described here:

<http://www.ixiacom.com/products/ixnetwork>

<http://www.ixiacom.com/products/ixvm>

# Reference architecture

The architecture of the test bed is shown in figure 1 below.

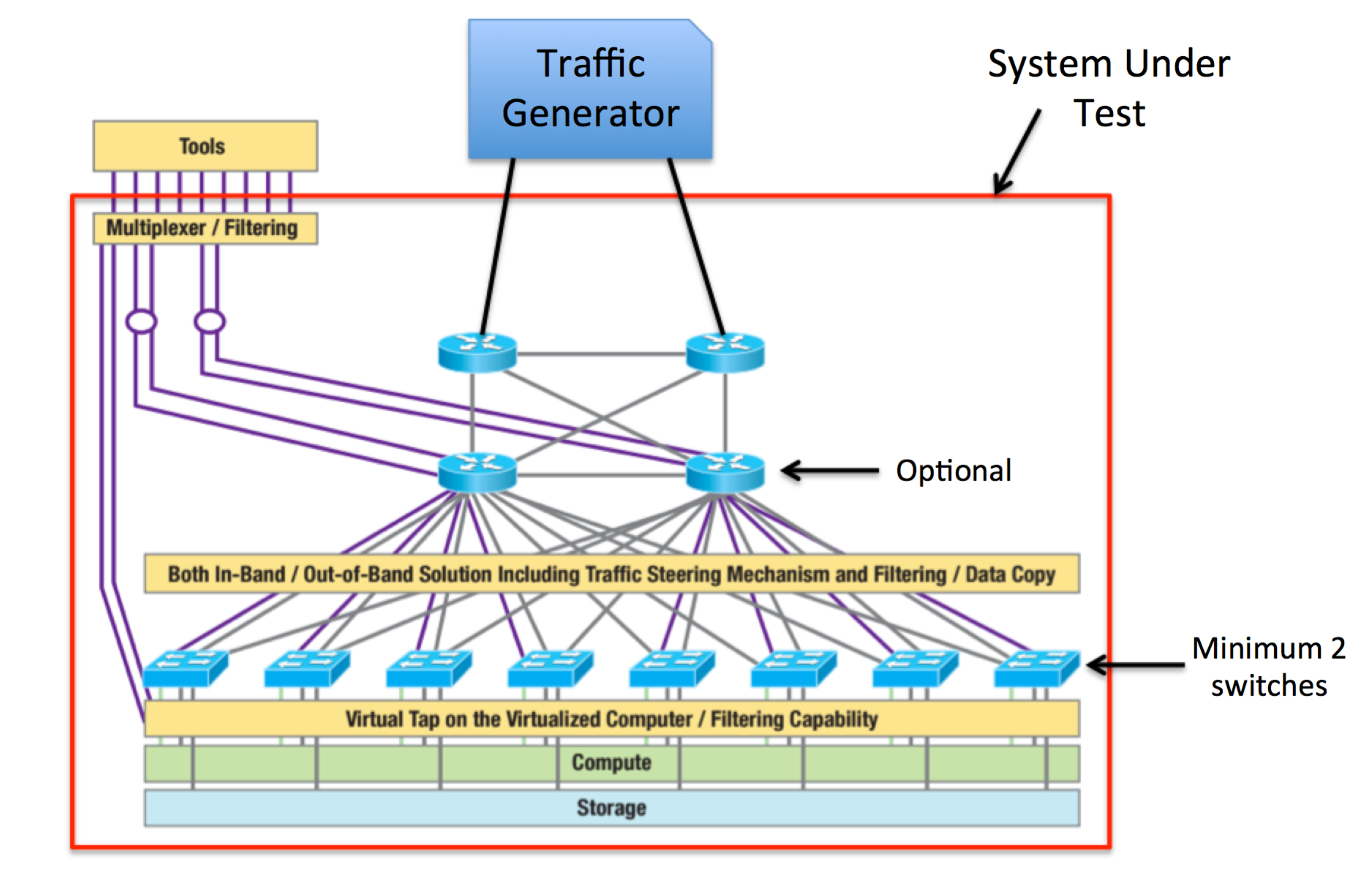


Figure 1. Reference architecture for tests

It shows the reference (or desired) architecture from the working group white paper, with a test tool connected to the network. Some notes:

* While the routers and switches are shown as being part of the System Under Test (SUT), they are not actually devices that are being tested with this plan. They are necessary components for the network that will be monitored by the real products under test
* Only two routers are necessary, and they should not be interconnected. The architecture is such that traffic must pass through the switches when transmitted from one port of the test tool, to the other port of the test tool
* The test tool will have two ports, one connected to each router. These ports will be the origin and termination of the test traffic
* Only two switches are necessary for the tests
* The test tool will be SW only
* Since these are functionality tests, the bandwidth of the interconnection ports are not relevant (1G, 10G, etc)

# Tests

## Requirement 1: Commodity hardware based upon merchant silicon with either an open or proprietary Switch OS

### Validation conditions (Pass)

Demonstrate (via documentation)

* The product is based on commodity HW
* The product OS used

## Requirement 2: Granular filtering based on 5-tuple and/or even more, including application signatures, and Quality of Service (QoS) marking capability.

### Test objective

To demonstrate the ability of the tools to filter based on IP addresses, port numbers, and DSCP value.

### Test architecture

The basic architecture from figure X will be used.

### Prerequisites

100 IP addresses reserved for the test

### Simulated traffic and procedure

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E

Procedure:

1. For each of the following steps, run the traffic for 30 seconds, and stop the traffic. Set up the filtering on the SUT before starting the traffic.
2. Demonstrate the ability to filter ingress and egress traffic for IP address x.x.x.x, and port number X
3. Demonstrate the ability to filter traffic between IP addresses z.z.z.z and y.y.y.y
4. Demonstrate the ability to filter all traffic with DSCP value = A

### Expected results

In each case above, the only traffic appearing towards the tools should be the filtered traffic

### Validation conditions (Pass)

Using a tool such as TCPdump, demonstrate that only the filtered traffic appears at the tools interface. No packet loss should be observed: packet counts from TCPdump should match those of the traffic generation tool for each filtered flow.

## Requirement 3: Capability to work with both underlay and overlay protocols, providing independent filtering on either, and/or correlate both traffic.

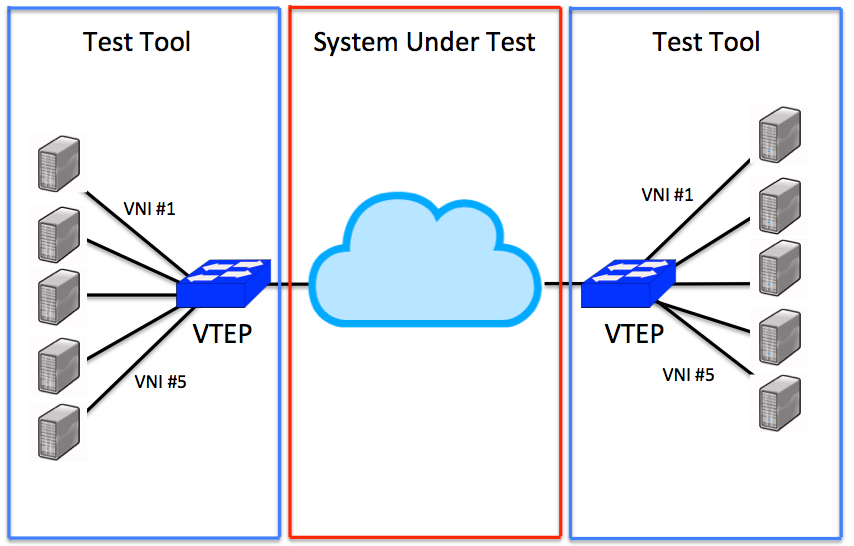
### Test objective

To demonstrate the ability to filter specific traffic encapsulated in an overlay tunnel.

### Test architecture

The basic architecture from figure X will be used.

The following diagram illustrates the VXLAN simulation



### Prerequisites

112 IP addresses

### Simulated traffic and procedure

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E
* 5 VXLAN based overlays (5 VNIs), using 2 VTEPs, and 10 IPs (5x5)
  + VTEPs and endpoints are simulated by the test tool
  + Traffic profile is identical to above

Procedure:

* Set a filter to select the traffic for one of the 5 VNIs
* Start the traffic
* Run traffic for 15 seconds
* Stop the traffic

### Expected results

The traffic from the filtered VNI should be visible at the tools interface

### Validation conditions (Pass)

* Compare the packet counts from the output data (tools interface) and the traffic generator for the filtered VNI. They should match

## Requirement 4: Process data without impact to production flow/processing flow (CPU/Memory/Bandwidth)

### Test objective

TBD. Performance test

### Test architecture

### Prerequisites

### Simulated traffic and procedure

### Expected results

### Validation conditions (Pass)

## Requirement 5: Horizontal scalability with the capability of resource management feedback.

### Test objective

TBD. Performance test

### Test architecture

### Prerequisites

### Simulated traffic and procedure

### Expected results

### Validation conditions (Pass)

## Requirement 6: Must be able to locally process data and create traffic signaling/alerting, while executing defined traffic based actions.

### Test objective

To demonstrate the ability to set a trigger on the traffic, and have an event to reflect that the trigger condition was met. The event can be an SNMP message, an email, syslog, etc.

### Test architecture

The basic architecture from figure X will be used.

### Prerequisites

102 IP addresses reserved for the test

### Simulated traffic and procedure

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E
* One additional flow, between IPs 101 and 102, using port Z, DSCP A

Procedure:

1. Set a filter for traffic between IP addresses x.x.x.x and y.y.y.y (one flow)
2. Set a trigger for traffic to IP address 102, port Z
3. Start the 50x50 meshed traffic
4. After 15 seconds, start the flow between IPs 101 and 102

### Expected results

An event reflecting that the trigger condition was met

### Validation conditions (Pass)

Demonstrate the presence of the event triggered by the network condition

## Requirement 7: Interoperability between vendors: integration and output that will support data collection integration. Open API for access and management (in/out).

### Test objective

Demonstrate that the filtered packet output of the SUT is correctly interpreted and consumed by at least two different manufacturers’ tools.

### Test architecture

The basic architecture from figure X will be used.

### Prerequisites

100 IP addresses reserved for the test

At least 2 output data consumer tools, from different manufacturers/providers

### Simulated traffic and procedure

The test is repeated once for each data consumer tool to be used to demonstrate interoperability.

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E

Procedure:

1. Set a filter for traffic having DSCP value B
2. Start traffic
3. Run traffic for 10 seconds
4. Stop traffic

### Expected results

That the data output consumer tools correctly reflect and display the filtered IP packets that were filtered and sent to them

### Validation conditions (Pass)

* Compare the packet counts between the data consumer tool and the traffic generator. They must match
* Demonstrate that the data consumer tool can display the set of IP packets received correctly (ie. The packets do not seem to be malformed, contain errors, etc)

## Requirement 8: Capability of Packet De-duplication/Packet Slicing/Data Masking and Application Recognition, including Packet Organization.

### Test objective

To demonstrate:

* The ability to mask packet contents
* The ability to slice packet contents
* The ability to remove duplicate packets from the output steered towards the tools

### Prerequisites

110 IP addresses reserved for the test

### Simulated traffic and procedure

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E
* 5 unidirectional flows between 10 IPs. Each flow is transmitting duplicate, identical packets to the destination at a rate of 10 pkts/sec.

Procedure:

1. Set a rule: all packets destined for IP x.x.x.x and port X to have 20 octets masked, starting at offset 100 from byte 0 of the IP header
2. Set a rule: all packets destined for IP y.y.y.y and port X to have 15 octets sliced, starting at offset 130 from byte 0 of the IP header
3. Start the main traffic
4. After 5 seconds, run the 5 individual flows containing packet duplicates for 1 second, then stop the flows
   1. Repeat this 1 second run 5 time, at 10 second intervals
5. Stop all traffic

### Expected results

* All required packets contents are masked at the appropriate location in the packet
* All required packets contents are masked at the appropriate location in the packet
* < 5% of total generated duplicates found in the output steered towards the tools

### Validation conditions (Pass)

Using tools such as TCPdump and Wireshark, verify that:

* All packets destined for IP x.x.x.x, port X have 20 octets masked from the original content, starting at offset 0 of the IP header
* All packets destined for IP y.y.y.y, port X have 15 octets removed, starting at offset 130 from the IP header
* Verify that less than 5% of the total duplicate packets sent have been forwarded to the tools interface

## Requirement 9: The ability to be Security and Compliance-aware.

### Test objective

To demonstrate the ability to recognize packets destined towards a (simulated) credit card processing server, and strip the payload from the packets in order to only deliver the headers of those packets towards the tools

### Test architecture

The basic architecture from figure X will be used.

### Prerequisites

100 IP addresses reserved for the test

At least 2 output data consumer tools, from different manufacturers/providers

### Simulated traffic and procedure

The test is repeated once for each data consumer tool to be used to demonstrate interoperability.

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E

Procedure:

1. Set a filter for traffic having DSCP value B
2. Set a rule such that traffic destined towards IP x.x.x.x for port X shall be interpreted as traffic for the credit card processor. Its packets are set to be stripped of the payload, and only the IP/UDP headers will be forwarded to the tools
3. Start traffic
4. Run traffic for 10 seconds
5. Stop traffic

### Expected results

All packets with destination IP x.x.x.x and port X with DSCP B shall be present and have only their headers visible.

### Validation conditions (Pass)

* Compare the packet counts between the data consumer tool and the traffic generator, for packets with destination IP x.x.x.x, port X and DSCP B. They must match
* All packets with destination IP x.x.x.x, port X and DSCP B must have only their headers (IP/UDP) present, with no payload
* All packets with destination IP x.x.x.x, port X and DSCP B must be readable by the consumer tool (or Wireshark) without errors

## Requirement 10: Multilayer visibility between underlay and overlay protocol use for management/Service-Level Agreement (SLA), monitoring/alerting, troubleshooting and reporting capabilities.

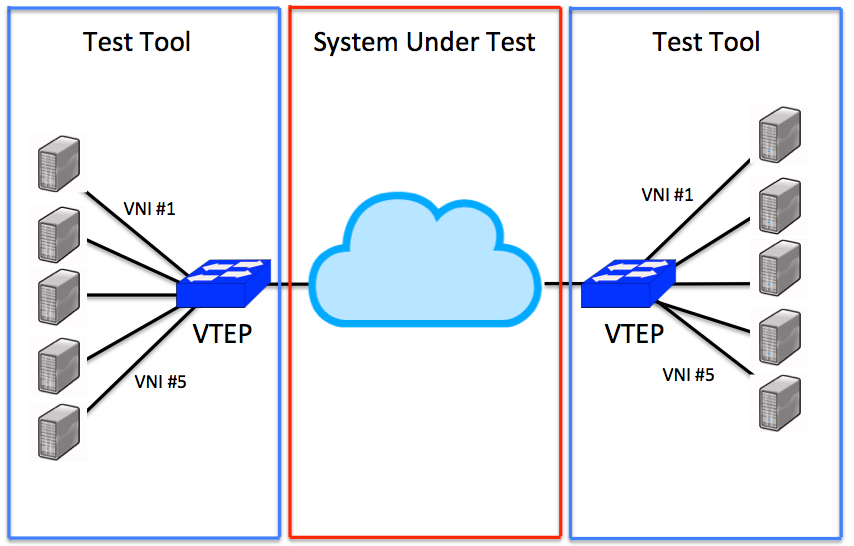
### Test objective

To demonstrate the ability to filter specific traffic encapsulated in an overlay tunnel.

### Test architecture

The basic architecture from figure X will be used.

The following diagram illustrates the VXLAN simulation



### Prerequisites

112 IP addresses

### Simulated traffic and procedure

The traffic will be:

* UDP traffic
* Fully meshed, 50 x 50 IPs (50 bi-directional flows)
* Packet size ranges between 250 and 1000 octets randomly
* Per flow packets per second: 50
* Each flow uses one of two source ports: X and Y
* Each flow uses one of 5 DSCP values: A, B, C, D or E
* 5 VXLAN based overlays (5 VNIs), using 2 VTEPs, and 10 IPs (5x5)
  + VTEPs and endpoints are simulated by the test tool
  + Traffic profile is identical to above

Procedure:

* Set a filter to select the traffic for one of the 5 VNIs
* Start the traffic
* Run traffic for 15 seconds
* Stop the traffic

### Expected results

The traffic from the filtered VNI should be visible at the tools interface

### Validation conditions (Pass)

* Compare the packet counts from the output data (tools interface) and the traffic generator for the filtered VNI. They should match