Internet-Draft
Network Working Group
Intended Status: Informational
Expires: August 22, 2015

Bhuvaneswaran Vengainathan
Anton Basil
Veryx Technologies
Mark Tassinari
Hewlett-Packard
Vishwas Manral
Ionos Corp
Sarah Banks
VSS Monitoring

March 23, 2015

# Benchmarking Methodology for SDN Controller Performance
# draft-bhuvan-bmwg-sdn-controller-benchmark-meth-00

## Abstract

This document defines the methodologies for benchmarking performance of SDN controllers. Terminology related to benchmarking SDN controller is described in the companion terminology document. SDN controllers have been implemented with many varying designs in order to achieve their intended network functionality. Hence, the authors have taken the approach of considering an SDN controller as a black box, defining the methodology in a manner that is agnostic to protocols and network services supported by controllers. The intent of this document is to provide a standard mechanism to measure the performance of all controller implementations.

## Status of this Memo

## Copyright Notice

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Scope . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3. Test Setup . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.1 Controller in Standalone Mode - Manage SDN Network . . . . 4
3.2 Controller in Cluster Mode - Manage SDN Network . . . . . . 4
3.3 Controller in Standalone Mode - Manage SDN Network with TP . 5
3.4 Controller in Cluster Mode - Manage SDN Network with TP . . 5
3.5 Controller in Standalone Mode - Manage SDN Node with TP . . 6
3.6 Controller in Cluster Mode - Manage SDN Node with TP . . . 7
4. Test Considerations . . . . . . . . . . . . . . . . . . . . 7
4.1 Network Topology . . . . . . . . . . . . . . . . . . . . . 7
4.2 Test Traffic . . . . . . . . . . . . . . . . . . . . . . . 7
4.3 Connection Setup . . . . . . . . . . . . . . . . . . . . . 8
4.4 Measurement Point Specification and Recommendation . . . . . 8
4.5 Connectivity Recommendation . . . . . . . . . . . . . . . . 8
5. Test Reporting . . . . . . . . . . . . . . . . . . . . . . . 9
6. Benchmarking Tests . . . . . . . . . . . . . . . . . . . . . 9
6.1 Performance . . . . . . . . . . . . . . . . . . . . . . . . 9
6.1.1 Network Topology Discovery Time . . . . . . . . . . . . . 9
6.1.2 Asynchronous Message Processing Time . . . . . . . . . . . 11
6.1.3 Asynchronous Message Processing Rate . . . . . . . . . . . 12
6.1.4 Path Provisioning Time . . . . . . . . . . . . . . . . . . 14
6.1.5 Path Provisioning Rate . . . . . . . . . . . . . . . . . . 16
6.1.6 Network Topology Change Detection Time . . . . . . . . . . 17
6.2 Scalability . . . . . . . . . . . . . . . . . . . . . . . . 18
6.2.1 Control Sessions Capacity. . . . . . . . . . . . . . . . . 18
6.2.2 Network Discovery Size . . . . . . . . . . . . . . . . . . 19
6.2.3 Forwarding Table Capacity . . . . . . . . . . . . . . . . 20
6.3 Security . . . . . . . . . . . . . . . . . . . . . . . . . 21
6.3.1 Exception Handling . . . . . . . . . . . . . . . . . . . . 21
6.3.2 Denial of Service Handling . . . . . . . . . . . . . . . . 22
6.4 Reliability . . . . . . . . . . . . . . . . . . . . . . . . 23
6.4.1 Controller Failover Time . . . . . . . . . . . . . . . . . 23
6.4.2 Network Re-Provisioning Time . . . . . . . . . . . . . . . 24

7. References . . . . . . . . . . . . . . . . . . . . . . . . 26
7.1 Normative References . . . . . . . . . . . . . . . . . . . . 26
7.2 Informative References . . . . . . . . . . . . . . . . . . . 26
8. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 26
9. Security Considerations . . . . . . . . . . . . . . . . . . . 27
10. Appendix A - Benchmarking Methodology using OF Controllers . 27
11. Acknowledgements . . . . . . . . . . . . . . . . . . . . . . 43
12. Authors' Addresses . . . . . . . . . . . . . . . . . . . . . 43

## 1. Introduction

This document provides generic methodologies for benchmarking SDN controller performance. An SDN controller may support many northbound and southbound protocols, implement a wide range of applications, and work solely, or as a group to achieve the desired functionality. This document considers an SDN controller as a black box, regardless of design and implementation. The tests defined in the document can be used to benchmark SDN controller for performance, scalability, reliability and security independent of northbound and southbound protocols. These tests can be performed on an SDN controller running as a virtual machine (VM) instance or on a bare metal server.  This document is intended for those who want to measure the SDN controller performance as well as compare various SDN controllers performance.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

## 2. Scope

This document defines methodology to measure the networking metrics of SDN controllers. The tests defined in this document enable benchmarking of both as a standalone and as a cluster of homogeneous controllers. These tests are recommended for execution in lab environments rather than in real time deployments. Performance benchmarking of federation of controllers is beyond the scope of this document.

## 3. Test Setup

The tests defined in this document enable measurement of SDN controller's performance in Standalone mode and Cluster mode. This section defines common reference topologies that are later referred to in individual tests.

## 3.1  Controller in Standalone Mode - Manage SDN Network

```
                           --------------------
                          |  SDN Applications  |
                           --------------------
                                     |
                                     | (Northbound interface)
                          -----------------------
                         |     SDN Controller    |
                         |         (DUT)         |
                          -----------------------
                                     | (Southbound interface)
                                     |
                   -------------------------------------
                  |                  |                  |
              ----------        ----------          ----------
             |   SDN    |  l1  |   SDN    |  ln-1  |   SDN    |
             |  Node 1  |------|  Node 2  |--..----|  Node n  |
              ----------        ----------          ----------
```

Figure 1

## 3.2 Controller in Cluster Mode - Manage SDN Network

```
                           --------------------
                          |  SDN Applications  |
                           --------------------
                                     |
                                     | (Northbound interface)
         ---------------------------------------------------------
        |  ------------------             ------------------      |
        | | SDN Controller 1 | <--E/W--> | SDN Controller n |      |
        |  ------------------             ------------------      |
         ---------------------------------------------------------
                                     | (Southbound interface)
                                     |
                   -------------------------------------
                  |                  |                  |
              ----------        ----------          ----------
             |   SDN    |  l1  |   SDN    |  ln-1  |   SDN    |
             |  Node 1  |------|  Node 2  |--..----|  Node n  |
              ----------        ----------          ----------
```

Figure 2

### 3.3 Controller in Standalone Mode - Manage SDN Network with Traffic Endpoints (TP)

```
                          --------------------
                         |  SDN Applications  |
                          --------------------
                                   |
                                   | (Northbound interface)
                         -----------------------
                        |  SDN Controller (DUT) |
                         -----------------------
                                     | (Southbound interface)
                                     |
                   -------------------------------------
                  |                  |                  |
              ----------        ----------          ----------
             |   SDN    |  l1  |   SDN    |  ln-1  |   SDN    |
             |  Node 1  |------|  Node 2  |--..----|  Node n  |
              ----------        ----------          ----------
                  |                                     |
                  | l0                                  | ln
                  |                                     |
            --------------                        --------------
           |   Traffic    |                      |   Traffic    |
           | Endpoint TP1 |                      | Endpoint TP2 |
            --------------                        --------------
```

Figure 3

### 3.4 Controller in Cluster Mode - Manage SDN Network with Traffic Endpoints (TP)

```
                          --------------------
                         |  SDN Applications  |
                          --------------------
                                     |
                                     | (Northbound interface)
        ---------------------------------------------------------
       |  ------------------             ------------------      |
       | | SDN Controller 1 | <--E/W--> | SDN Controller n |     |
       |  ------------------             ------------------      |
        ---------------------------------------------------------
                                     |
```

```
                                     | (Southbound interface)
                                     |
                   -------------------------------------
                  |                  |                  |
              ----------        ----------          ----------
             |   SDN    |  l1  |   SDN    |  ln-1  |   SDN    |
             |  Node 1  |------|  Node 2  |--..----|  Node n  |
              ----------        ----------          ----------
                  |                                     |
                  | l0                                  | ln
                  |                                     |
            --------------                        --------------
           |   Traffic    |                      |   Traffic    |
           | Endpoint TP1 |                      | Endpoint TP2 |
            --------------                        --------------

                                  Figure 4
```

### 3.5 Controller in Standalone Mode - Manage SDN Node with Traffic Endpoints (TP)

```
                          --------------------
                         |  SDN Applications  |
                          --------------------
                                   |
                                   | (Northbound interface)
                         -----------------------
                        |     SDN Controller    |
                        |         (DUT)         |
                         -----------------------
                                   | (Southbound interface)
                                   |
                         l0    ----------   l1
                       -------|   SDN    |---------
                      |       |  Node 1  |         |
                      |        ----------          |
                  ----------                  ----------
                 | Traffic  |                | Traffic  |
                 | Endpoint |                | Endpoint |
                 |   TP1    |                |   TP2    |
                  ----------                  ----------

                                  Figure 5
```

### 3.6 Controller in Cluster Mode - Manage SDN Node with Traffic Endpoints (TP)

```
                          --------------------
                         |  SDN Applications  |
                          --------------------
                                   |
                                   | (Northbound interface)
        ---------------------------------------------------------
       |  ------------------             ------------------      |
       | | SDN Controller 1 | <--E/W--> | SDN Controller n |     |
       |  ------------------             ------------------      |
        ---------------------------------------------------------
                                   | (Southbound interface)
                                   |
                          l0   ----------     l1
                       -------|   SDN    |---------
                      |       |  Node 1  |         |
                      |        ----------          |
                  ----------                  ----------
                 | Traffic  |                | Traffic  |
                 | Endpoint |                | Endpoint |
                 |   TP1    |                |   TP2    |
                  ----------                  ----------
```

Figure 6

## 4. Test Considerations

### 4.1 Network Topology

The network SHOULD be deployed with SDN nodes interconnected in an either fully meshed, tree or linear topology. Care should be taken to make sure that a loop prevention mechanism is enabled either in the SDN controller, or in the network. To achieve a complete performance characterization of the SDN controller, it is recommended that the controller be benchmarked for many network topologies. These network topologies can be deployed using real hardware or emulated in hardware platforms.

### 4.2 Test Traffic

Test traffic can be used to notify the controller about the arrival of new flows or generate notifications/events towards controller. In either case, it is recommended to use multiple frame sizes as recommended in RFC 2544 for benchmarking.

### 4.3 Connection Setup

There may be controller implementations that support unencrypted and encrypted network connections with SDN nodes. Further, the controller may have backward compatibility with SDN nodes running older versions of southbound protocols. It is recommended that the controller performance be measured with one or more applicable connection setup methods defined below.

1. Unencrypted connection with SDN nodes, running same protocol version.
2. Unencrypted connection with SDN nodes, running different protocol versions.
   Example:
   1. Controller running current protocol version and switch running older protocol version
   2. Controller running older protocol version and switch running current protocol version
3. Encrypted connection with SDN nodes, running same protocol version
4. Encrypted connection with SDN nodes, running different protocol versions.
   Example:
   1. Controller running current protocol version and switch running older protocol version
   2. Controller running older protocol version and switch running current protocol version

### 4.4 Measurement Point Specification and Recommendation

The measurement accuracy depends on several factors including the point of observation where the indications are captured. For example, the notification can be observed at the ingress or egress point of the SDN node. If it is observed at the egress point of the SDN node, the measurement includes the latency within the SDN node also. It is recommended to make observation at the ingress point of the SDN node unless it is explicitly mentioned otherwise in the individual test.

### 4.5 Connectivity Recommendation

SDN controller in the test setup may be connected with the deployed SDN nodes over shared link with aggregation points. In this case it is recommended to ensure that the intermediate devices does not introduce any delays or fail during benchmarking tests.

## 5. Test Reporting

Each test has a reporting format which is specific to individual tests. In addition, the following configuration parameters MUST be reflected in the test report.

1. Controller name and version
2. Northbound protocols and versions
3. Southbound protocols and versions
4. Controller redundancy mode (Standalone or Cluster Mode)
5. Connection setup (Unencrypted or Encrypted)
6. Network Topology (Mesh or Tree or Linear)
7. SDN Node Type (Physical or Virtual or Emulated)
8. Number of Nodes
9. Number of Links
10. Test Traffic Type
11. Controller System Configuration (e.g., CPU, Memory, Operating System, Interface Speed etc.,)
12. Reference Test Setup (e.g., Section 3.1 etc.,)

## 6. Benchmarking Tests

### 6.1 Performance

#### 6.1.1 Network Topology Discovery Time

Objective:

To measure the time taken to discover the network topology - nodes and links by a controller, expressed in milliseconds.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. The controller MUST support network discovery.
2. Tester should be able to retrieve the discovered topology information either through controller's management interface or northbound interface to determine if the discovery was successful and complete.
3. Ensure that the controller's topology re-discovery timeout has been set to the maximum value to avoid initiation of re-discovery process in the middle of the test.
4. SDN nodes in the network MUST have zero backoff interval.

Procedure:

1. Initialize the controller - network applications, northbound and southbound interfaces.
2. Deploy the network with the given number of nodes using mesh or linear topology.
3. Initialize the network connections between controller and network nodes.
4. Record the time for the first discovery message sent by the controller to the SDN nodes (Tm1).
5. Query the controller for every 3 seconds to obtain the discovered network topology information through northbound interface or management interface and compare it with the deployed network topology information.
6. Stop the test when the discovered topology information is matching with the deployed network topology or the discovered topology information for 3 consecutive queries return the same
7. Record the time last discovery message sent by the controller to the SDN nodes (Tmn) when the test completed successfully. (e.g., the topology matches).

Note:

1. While recording the Tmn value, it is recommended that the messages that are used for aliveness checks or session management be ignored.

Measurement:

Topology Discovery Time Tr1 = Tmn-Tm1.

```
                                  Tr1 + Tr2 + Tr3 .. Trn
Average Topology Discovery Time = -----------------------
                                  Total Test Iterations
```

Note:

1. The measurement inherently includes the network latency introduced by the SDN nodes and the link connecting them. To minimize the impact of network latency on this test, it is recommended to consider the following
   a. Perform the test without any network traffic other than the test traffic.
   b. Minimize the number of network components connecting SDN nodes and the controller.
2. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.
3. To get the full characterization of a controller's topology discovery functionality
   a. Perform the test with varying number of nodes using same topology

b. Perform the test with same number of nodes using different topologies.

Reporting Format:

The Topology Discovery Time results MUST be reported in the format of a table, with a row for each successful iteration. The last row of the table indicates the average Topology Discovery Time.

If this test is repeated with varying number of nodes over the same topology, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Number of nodes (N), the Y coordinate SHOULD be the average Topology Discovery Time.

If this test is repeated with same number of nodes over different topologies, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Topology Type, the Y coordinate SHOULD be the average Topology Discovery Time.

### 6.1.2 Asynchronous Message Processing Time

Objective:

To measure the time taken by the controller to process a asynchronous message, expressed in milliseconds.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. The controller MUST have completed the network topology discovery for the connected nodes.

Procedure:

1. Generate asynchronous messages from every connected nodes one at a time for the test duration.
2. Record every request transmit (T1) timestamp and the corresponding response (R1) received timestamp for every successful message exchange.

Measurement:

$$\text{Asynchronous Message Processing Time Tr1} = \frac{(R1-T1) + (R2-T2)..(Rn-Tn)}{Nrx}$$

Where Nrx is the total number of successful messages exchanged

$$\text{Average Asynchronous Message Processing Time} = \frac{Tr1 + Tr2 + Tr3..Trn}{\text{Total Test Iterations}}$$

Note:

1. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.
2. To get the full characterization of a controller's asynchronous message processing time
   a. Perform the test with varying number of nodes using same topology
   b. Perform the test with same number of nodes using different topologies.

Reporting Format:

The Asynchronous Message Processing Time results MUST be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Asynchronous Message Processing Time.

The report should capture the following information in addition to the configuration parameters captured in section 5.
- Successful messages exchanged (Nrx)

If this test is repeated with varying number of nodes with same topology, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Number of nodes (N), the Y coordinate SHOULD be the average Asynchronous Message Processing Time.

If this test is repeated with same number of nodes using different topologies, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Topology Type, the Y coordinate SHOULD be the average Asynchronous Message Processing Time.

### 6.1.3 Asynchronous Message Processing Rate

Objective:

To measure the maximum number of asynchronous messages (session aliveness check message, new flow arrival notification message etc.) a controller can process within the test duration, expressed in messages processed per second.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. The controller MUST have completed the network topology discovery for the connected nodes.
2. Ensure that the intermediate devices are provisioned such that it does not drop the messages sent to the controller.

Procedure:

1. Generate asynchronous messages continuously from all the connected nodes for the Test Duration (Td).
2. Record total number of responses received from the controller (Nrx) as well as the number of messages sent(Ntx) to the controller from all SDN nodes within the test duration(Td).

Measurement:

$$\text{Asynchronous Message Processing Rate Tr1} = \frac{Nrx}{Td}$$

$$\text{Average Asynchronous Message Processing Rate} = \frac{Tr1 + Tr2 + Tr3..Trn}{\text{Total Test Iterations}}$$

Loss Ratio = (Ntx-Nrx)/100.

Note:

1. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.
2. To get the full characterization of a controller's asynchronous message processing rate
   a. Perform the test with varying number of nodes using same topology.
   b. Perform the test with same number of nodes using different topologies.

Reporting Format:

The Asynchronous Message Processing Rate results MUST be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Asynchronous Message Processing Rate.

The report should capture the following information in addition to the configuration parameters captured in section 5.

- Offered rate (Ntx)
- Loss Ratio

If this test is repeated with varying number of nodes over same topology, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Number of nodes (N), the Y coordinate SHOULD be the average Asynchronous Message Processing Rate.

If this test is repeated with same number of nodes over different topologies, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Topology Type, the Y coordinate SHOULD be the average Asynchronous Message Processing Rate.

### 6.1.4 Path Provisioning Time

Objective:

To measure the time taken by the controller to setup a path between source and destination node, expressed in milliseconds.

Reference Test Setup:

The test can use one of the test setups described in section 3.3 and section 3.4 of this document.

Prerequisite:

1. The controller MUST contain the network topology information for the deployed network topology.
2. The network topology information can be learnt through dynamic Topology Discovery Mechanism or static configuration.
3. The controller should have the knowledge about the location of destination endpoint for which the path has to be provisioned. This can be achieved through dynamic learning or static provisioning.
4. Ensure that the default action for flow miss in SDN node is 'send to controller'.

Procedure:

Reactive Flow Provisioning Mode:

1. Send traffic with source as source endpoint address and destination as destination endpoint address from TP1.
2. Wait for the arrival of first frame from the destination node or the expiry of test duration (Td).
3. Record the time of the first flow provisioning request message sent to the controller(Tsf1).
4. Record the time of the last flow provisioning response message received from the controller(Tdf1).

Proactive Flow Provisioning Mode:

1. Send traffic with source as source endpoint address and destination as destination endpoint address from TP1.
2. Install the flow entries to reach from source endpoint to the destination endpoint through controller's northbound or management interface.
3. Wait for the arrival of first frame from the destination node or the expiry of test duration (Td).
4. Record the time when proactive flow is provisioned in the Controller (Tsf1).
5. Record the time of the last flow provisioning message received from the controller(Tdf1).

Measurement:

Flow Provisioning Time Tr1 = Tdf1-Tsf1.

$$\text{Average Path Provisioning Time} = \frac{Tr1 + Tr2 + Tr3 \text{ .. } Trn}{\text{Total Test Iterations}}$$

Note:

1. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.
2. To get the full characterization of a controller's path provisioning time
   a. Perform the test with varying number of nodes using same topology
   b. Perform the test with same number of nodes using different topologies.

Reporting Format:

The Path Provisioning Time results MUST be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Path Provisioning Time.

The report should capture the following information in addition to the configuration parameters captured in section 5.

- Number of data path nodes

If this test is repeated with varying number of nodes with same topology, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Number of nodes (N), the Y coordinate SHOULD be the average Path Provisioning Time.

If this test is repeated with same number of nodes using different topologies, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Topology Type, the

Y coordinate SHOULD be the average Path Provisioning Time.

### 6.1.5 Path Provisioning Rate

Objective:

To measure the maximum number of independent paths a controller can concurrently establish between source and destination nodes within the test duration, expressed in paths per second.

Reference Test Setup:

The test can use one of the test setup described in section 3.3 and section 3.4 of this document.

Prerequisite:

1. The controller MUST contain the network topology information for the deployed network topology.
2. The network topology information can be learnt through dynamic Topology Discovery Mechanism or static configuration.
3. The controller should have the knowledge about the location of destination endpoints for which the paths have to be provisioned. This can be achieved through dynamic learning or static provisioning.
4. Ensure that the default action for flow miss in SDN node is 'send to controller'.

Procedure:

Reactive Flow Provisioning Mode:

1. Send traffic at the individual node's asynchronous message processing rate with unique source and destination addresses from test port TP1.
2. Record total number of unique frames received from the destination node (Ndf) within the test duration (Td).

Proactive Flow Provisioning Mode:

1. Send traffic continuously with unique source and destination addresses from test port TP1.
2. Install corresponding flow entries to reach from source endpoints to the destination endpoint through controller's northbound or management interface.
3. Record total number of unique frames received from the destination node (Ndf) within the test duration (Td).

Measurement:

$$\text{Path Provisioning Rate Tr1} = \frac{\text{Ndf}}{\text{Td}}$$

$$\text{Average Path Provisioning Rate} = \frac{\text{Tr1 + Tr2 + Tr3 .. Trn}}{\text{Total Test Iterations}}$$

Note:

1. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.
2. To get the full characterization of a controller's path provisioning rate
   a. Perform the test with varying number of nodes using same topology
   b. Perform the test with same number of nodes using different topologies.

Reporting Format:

The Path Provisioning Rate results MUST be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Path Provisioning Rate.

The report should capture the following information in addition to the configuration parameters captured in section 5.

- Number of Nodes in the path
- Provisioning Type (Proactive/Reactive)
- Offered rate

If this test is repeated with varying number of nodes with same topology, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Number of nodes (N), the Y coordinate SHOULD be the average Path Provisioning Rate.

If this test is repeated with same number of nodes using different topologies, the results SHOULD be reported in the form of a graph. The X coordinate SHOULD be the Topology Type, the Y coordinate SHOULD be the average Path Provisioning Rate.

### 6.1.6 Network Topology Change Detection Time

Objective:

To measure the time taken by the controller to detect any changes in the network topology, expressed in milliseconds.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. The controller MUST have discovered the network topology information for the deployed network topology.
2. The periodic network discovery operation should be configured to twice the Test duration (Td) value.

Procedure:

1. Trigger a topology change event through one of the operation (e.g., Add a new node or bring down an existing node or a link).
2. Record the time when the first topology change notification is sent to the controller (Tcn).
3. Stop the test when the controller sends the first topology re-discovery message to the SDN node or the expiry of test interval (To).
4. Record the time when the first topology re-discovery message is received from the controller (Tcd).

Measurement:

Network Topology Change Detection Time Tr1 = Tcd-Tcn.

```
                                  Tr1 + Tr2 + Tr3 .. Trn
Average Network Topology Change
                Detection Time = ---------------------------
                                  Total Test Iterations
```

Note:

1. To increase the confidence in measured result, it is recommended that this test be performed several times with same number of nodes using same topology.

Reporting Format:

The Network Topology Change Detection Time results MUST be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Network Topology Change Time.

## 6.2 Scalability

### 6.2.1 Control Session Capacity

Objective:

To measure the maximum number of control sessions the controller can maintain.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. Ensure SDN nodes have no inter-node links.

Procedure:

1. Initialize control connection with controller from every SDN node in the network
2. Stop the test when the controller starts dropping the control connection.
3. Record the number of successful connections established with the controller (CCn).

Measurement:

Control Sessions Capacity = CCn.

Reporting Format:

The Control Session Capacity results MUST be reported in addition to the configuration parameters captured in section 5.

### 6.2.2 Network Discovery Size

Objective:

To measure the network size (number of nodes, links and hosts) that a controller can discover.

Reference Test Setup:

The test can use one of the test setup described in section 3.1 and section 3.2 of this document.

Prerequisite:

1. The controller MUST support automatic network discovery.
2. Tester should be able to retrieve the discovered topology information either through controller's management interface or northbound interface.
3. Controller should be operational.
4. Network with the given number of nodes and intended topology (Mesh or Linear or Tree) should be deployed.

Procedure:

1. Initialize the network connections between controller and network nodes.
2. Query the controller for the discovered network topology information and compare it with the deployed network topology information.

3a. Increase the number of nodes by 1 when the comparison is successful and repeat the test.

3b. Decrease the number of nodes by 1 when the comparison fails and repeat the test.

4. Continue the test until the comparison of step 3b is successful.
5. Record the number of nodes for the last iteration (Ns) where

the topology comparison was successful.

Measurement:

Network Discovery Size = Ns.

Note:

This test may be performed with different topologies to obtain the controller's scalability factor for various network topologies.

Reporting Format:

The Network Discovery Size results MUST be reported in addition to the configuration parameters captured in section 5.

### 6.2.3 Forwarding Table Capacity

Objective:

To measure the maximum number of flow entries a controller can manage in its Forwarding table.

Reference Test Setup:

The test can use one of the test setups described in section 3.5 and section 3.6 of this document.

Prerequisite:

1. The controller Forwarding table should be empty.
2. Flow Idle time MUST be set to higher or infinite value.
3. The controller MUST have completed network topology discovery.
4. Tester should be able to retrieve the forwarding table information either through controller's management interface or northbound interface.

Procedure:

Reactive Flow Provisioning Mode:

1. Send bi-directional traffic continuously with unique source and/or destination addresses from test ports TP1 and TP2 at the asynchronous message processing rate of controller.
2. Query the controller at a regular interval (e.g., 5 seconds) for the number of flow entries from its northbound interface.
3. Stop the test when the retrieved value is constant for three consecutive iterations and record the value received from the last query (Nrp).

Proactive Flow Provisioning Mode:

1. Install unique flows continuously through controller's northbound or management interface until a failure response is received from the controller.
2. Record the total number of successful responses (Nrp).

Note:

Some controller designs for proactive flow provisioning mode may require the switch to send flow setup requests in order to generate flow setup responses. In such cases, it is recommended to generate bi-directional traffic for the provisioned flows.

Measurement:

Proactive Flow Provisioning Mode:

Max Flow Entries = Total number of flows provisioned (Nrp)

Reactive Flow Provisioning Mode:

Max Flow Entries = Total number of learnt flow entries (Nrp)

Forwarding Table Capacity = Max Flow Entries.

Reporting Format:

The Forwarding Table Capacity results MUST be tabulated with the following information in addition to the configuration parameters captured in section 5.

- Provisioning Type (Proactive/Reactive)

### 6.3 Security

#### 6.3.1 Exception Handling

Objective:

To determine the effect of handling error packets and notifications on performance tests. The impact MUST be measured for the following performance tests

a. Path Provisioning Rate<br>
b. Path Provisioning Time<br>
c. Network Topology Change Detection Time

Reference Test Setup:

The test can use one of the test setups described in section 3.5 and section 3.6 of this document.

Prerequisite:

1. This test MUST be performed after obtaining the baseline measurement results for the above performance tests.
2. Ensure that the invalid messages are not dropped by the intermediate devices connecting the controller and SDN nodes.

Procedure:

1. Perform the above listed performance tests and send 1% of messages from the Asynchronous Message Processing Rate as invalid messages from the connected nodes.
2. Perform the above listed performance tests and send 2% of messages from the Asynchronous Message Processing Rate as invalid messages from the connected nodes.

Note:

1. Invalid messages can be frames with incorrect protocol fields or any form of failure notifications sent towards controller.

Measurement:

Measurement MUST be done as per the equation defined in the corresponding performance test measurement section.

Reporting Format:

The Exception Handling results MUST be reported in the format of table with a column for each of the below parameters and row for each of the listed performance tests.
- Without Exceptions
- With 1% Exceptions
- With 2% Exceptions

### 6.3.2 Denial of Service Handling

Objective:

To determine the effect of handling DoS attacks on performance and scalability tests The impact MUST be measured for the following tests

a. Path Provisioning Rate
b. Path Provisioning Time
c. Network Topology Change Detection Time
d. Network Discovery Size

Reference Test Setup:

The test can use one of the test setups described in section 3.5 and section 3.6 of this document.

Prerequisite:

This test MUST be performed after obtaining the baseline measurement results for the above tests.

Procedure:

1. Perform the listed tests and launch DoS attack towards controller while the test is running.

Note:

DoS attacks can be launched on one of the following interfaces.

a. Northbound (e.g., Sending a huge number of requests on northbound interface)

b. Management (e.g., Ping requests to controller's management interface)

c. Southbound (e.g., TCP SYNC messages on southbound interface)

Measurement:

Measurement MUST be done as per the equation defined in the corresponding test's measurement section.

Reporting Format:

The DoS Attacks Handling results MUST be reported in the format of table with a column for each of the below parameters and row for each of the listed tests.

- Without any attacks
- With attacks

The report should also specify the nature of attack and the interface.

### 6.4 Reliability

#### 6.4.1 Controller Failover Time

Objective:

The time taken to switch from an active controller to the backup controller, when the controllers work in redundancy mode and the active controller fails.

Reference Test Setup:

The test can use the test setup described in section 3.4 of this document.

Prerequisite:

1. Master controller election MUST be completed.
2. Nodes are connected to the controller cluster as per the Redundancy Mode (RM).
3. The controller cluster should have completed the network topology discovery.
4. The SDN Node MUST send all new flows to the controller when it receives.
5. Controller should have learnt the location of destination (D1) at Test Port TP2.

Procedure:

1. Send uni-directional traffic continuously with incremental sequence number and source addresses from test ports TP1 at the rate that the controller processes without any drops.
2. Bring down the active controller.
3. Stop the test when a first frame received on TP2 after failover operation.
4. Record the time at which the last valid frame received (T1) at test port TP2 before sequence error and the first valid frame received (T2)after the sequence error at test port TP2.

Measurement:

Controller Failover Time = (T2 - T1)
Packet Loss = Number of missing packet sequences.

Note:

1. Ensure that there are no packet drops observed at the test port TP2 before bringing down the controller.

Reporting Format:

The Controller Failover Time results MUST be tabulated with the following information.

- Number of cluster nodes
- Redundancy mode
- Controller Failover
- Time Packet Loss
- Cluster keep-alive interval

### 6.4.2 Network Re-Provisioning Time

Objective:

To compute the time taken to re-route the traffic by the controller when there is a failure in existing traffic paths.

Reference Test Setup:

The test can use one of the test setup described in section 3.3 and section 3.4 of this document.

Prerequisite:

1. Network with the given number of nodes and intended topology (Mesh or Tree) with redundant paths MUST be deployed.
2. Ensure that the controller MUST have knowledge about the location of traffic endpoints TP1 and TP2.

Procedure:

1. Send bi-directional traffic continuously with unique sequence number from TP1 and TP2.
2. Bring down a link or switch in the traffic path.
3. Stop the test after receiving first frame after network re-convergence (timeline).
4. Record the time of last received frame prior to the frame loss at TP2 (TP2-Tlfr) and the time of first frame received after the frame loss at TP2 (TP2-Tffr).
5. Record the time of last received frame prior to the frame loss at TP1 (TP1-Tlfr) and the time of first frame received after the frame loss at TP1 (TP1-Tffr).

Note:

1. Ensure that the controller does not pre-provision the alternate path in the SDN nodes.
   Duplicate traffic check??

Measurement:

```
   Forward Direction Path Re-Provisioning Time (FDRT)
                                          = (TP2-Tffr - TP2-Tlfr)

   Reverse Direction Path Re-Provisioning Time (RDRT)
                                          =  (TP1-Tffr - TP1-Tlfr)
```

Network Re-Provisioning Time = (FDRT+RDRT)/2

Forward Direction Packet Loss = Number of missing sequence frames at TP1

Reverse Direction Packet Loss = Number of missing sequence frames at TP2

Reporting Format:

The Network Re-Provisioning Time results MUST be tabulated with the following information.

- Number of nodes in the primary path
- Number of nodes in the alternate path
- Network Re-Provisioning Time
- Forward Direction Packet Loss
- Reverse Direction Packet Loss

## 7. References

### 7.1 Normative References

[RFC2544] S. Bradner, J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices",RFC 2544, March 1999.

[RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics",RFC 2330, May 1998.

[RFC6241] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman, "Network Configuration Protocol (NETCONF)",RFC 6241, June 2011.

[RFC6020] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010

[RFC5440] JP. Vasseur, JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[OpenFlow Switch Specification] ONF,"OpenFlow Switch Specification" Version 1.4.0 (Wire Protocol 0x05), October 14, 2013.

[I-D.sdn-controller-benchmark-term] Bhuvaneswaran.V, Anton Basil, Mark.T, Vishwas Manral, Sarah Banks "Terminology for Benchmarking SDN Controller Performance", draft-bhuvan-bmwg-sdn-controller-benchmark-term-00 (Work in progress), March 23, 2015

[I-D.i2rs-architecture] A. Atlas, J. Halpern, S. Hares, D. Ward, T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-09 (Work in progress), March 6, 2015

### 7.2 Informative References

[OpenContrail] Ankur Singla, Bruno Rijsman, "OpenContrail Architecture Documentation", http://opencontrail.org/opencontrail-architecture-documentation

[OpenDaylight] OpenDaylight Controller:Architectural Framework, https://wiki.opendaylight.org/view/OpenDaylight_Controller

## 8. IANA Considerations

This document does not have any IANA requests.

## 9. Security Considerations

Benchmarking tests described in this document are limited to the performance characterization of controller in lab environment with isolated network and dedicated address space.

## 10. Appendix A - Benchmarking Methodology using OpenFlow(OF) Controllers

This section gives an overview of OpenFlow protocol and provides test methodology to benchmark SDN controllers supporting OpenFlow southbound protocol.

### 10.1. Protocol Overview

OpenFlow is an open standard protocol defined by Open Networking Foundation (ONF), used for programming the forwarding plane of network switches or routers via a centralized controller.

### 10.2. Messages Overview

OpenFlow protocol supports three messages types namely controller-to-switch, asynchronous and symmetric.

Controller-to-switch messages are initiated by the controller and used to directly manage or inspect the state of the switch. These messages allow controllers to query/configure the switch (Features, Configuration messages), collect information from switch (Read-State message), send packets on specified port of switch (Packet-out message), and modify switch forwarding plane and state (Modify-State, Role-Request messages etc.).

Asynchronous messages are generated by the switch without a controller soliciting them. These messages allow switches to update controllers to denote an arrival of new flow (Packet-in), switch state change (Flow-Removed, Port-status) and error (Error).

Symmetric messages are generated in either direction without solicitation. These messages allow switches and controllers to set up connection (Hello), verify for liveness (Echo) and offer additional functionalities (Experimenter).

### 10.3. Connection Overview

OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. The OpenFlow channel connection can be setup using plain TCP or TLS. By default, a switch establishes single connection with SDN controller. A switch may establish multiple parallel connections to single controller (auxiliary connection) or multiple controllers to handle controller failures and load balancing.

## 10.4 Performance Benchmarking Tests

### 10.4.1 Network Topology Discovery Time

Procedure:

```
      OpenFlow                    OpenFlow                     Tester
      Switches                   Controller
         |                            |                           |
         |                            |     <Initialize controller|
         |                            | app.,NB and SB interfaces>|
         |                            |                           |
         |                            |     <Deploy network with  |
         |                            |  given no. of OF switches>|
         |                            |                           |
         |    OFPT_HELLO Exchange     |                           |
         |<-------------------------->|                           |
         |                            |                           |
         |    PACKET_OUT with LLDP    |                           |
         |      to all switches       |                           |
    (Tm1)|<---------------------------|                           |
         |                            |                           |
         |         PACKET_IN with LLDP|                           |
         |          rcvd from switch-1|                           |
         |--------------------------->|                           |
         |                            |                           |
         |         PACKET_IN with LLDP|                           |
         |          rcvd from switch-2|                           |
         |--------------------------->|                           |
         |            .               |                           |
         |            .               |                           |
         |                            |                           |
         |         PACKET_IN with LLDP|                           |
         |          rcvd from switch-n|                           |
    (Tmn)|--------------------------->|                           |
         |                            |                           |
         |                            |      <Wait for the expiry |
         |                            |     of Test Duration (Td)>|
         |                            |                           |
         |                            |   Query the controller for|
         |                            |   discovered n/w topo.(Di)|
         |                            |<--------------------------|
         |                            |                           |
         |                            |   <Compare the discovered |
         |                            |    & offered n/w topology>|
         |                            |                           |
```

Legend:

NB: Northbound<br>
SB: Southbound<br>
OF: OpenFlow<br>
Tm1: Time of reception of first LLDP message from controller<br>
Tmn: Time of last LLDP message sent to controller

Discussion:

The Network Topology Discovery Time can be obtained by calculating the time difference between the first PACKET_OUT with LLDP message received from the controller (Tm1) and the last PACKET_IN with LLDP message sent to the controller (Tmn) when the comparison is successful.

### 10.4.2 Asynchronous Message Processing Time

Procedure:

```
      OpenFlow                    OpenFlow                      Tester
      Switches                   Controller
         |                            |                           |
         |PACKET_IN with single       |                           |
         |OFP match header            |                           |
     (T0)|--------------------------->|                           |
         |                            |                           |
         | PACKET_OUT with single OFP |                           |
         |              action header |                           |
     (R0)|<---------------------------|                           |
         |          .                 |                           |
         |          .                 |                           |
         |          .                 |                           |
         |                            |                           |
         |PACKET_IN with single OFP   |                           |
         |match header                |                           |
     (Tn)|--------------------------->|                           |
         |                            |                           |
         | PACKET_OUT with single OFP |                           |
         |               action header|                           |
     (Rn)|<---------------------------|                           |
         |                            |                           |
         |                            |    <Wait for the expiry of|
         |                            |           Test Duration>  |
         |                            |                           |
         |                            |     <Record the number of |
         |                            |    PACKET_INs/PACKET_OUTs |
         |                            |           Exchanged (Nrx)>|
         |                            |                           |
```

Legend:

T0,T1, ..Tn are PACKET_IN messages transmit timestamps.
R0,R1, ..Rn are PACKET_OUT messages receive timestamps.
Nrx : Number of successful PACKET_IN/PACKET_OUT message exchanges

Discussion:

The Asynchronous Message Processing Time will be obtained by sum of ((R0-T0),(R1-T1)..(Rn - Tn))/ Nrx.

### 10.4.3 Asynchronous Message Processing Rate

Procedure:

```
      OpenFlow                    OpenFlow                    Tester
      Switches                   Controller
         |                            |                           |
         |PACKET_IN with multiple OFP |                           |
         |match headers               |                           |
         |--------------------------->|                           |
         |                            |                           |
         | PACKET_OUT with multiple   |                           |
         |          OFP action headers|                           |
         |<---------------------------|                           |
         |                            |                           |
         |PACKET_IN with multiple OFP |                           |
         |match headers               |                           |
         |--------------------------->|                           |
         |                            |                           |
         | PACKET_OUT with multiple   |                           |
         |          OFP action headers|                           |
         |<---------------------------|                           |
         |            .               |                           |
         |            .               |                           |
         |            .               |                           |
         |                            |                           |
         |PACKET_IN with multiple OFP |                           |
         |match headers               |                           |
         |--------------------------->|                           |
         |                            |                           |
         | PACKET_OUT with multiple   |                           |
         |          OFP action headers|                           |
         |<---------------------------|                           |
         |                            |                           |
         |                            |    <Wait for the expiry of|
         |                            |           Test Duration>  |
         |                            |                           |
         |                            |  <Record the number of OFP|
         |                            |       action headers rcvd>|(Nrx)
         |                            |                           |
```

Discussion:

The Asynchronous Message Processing Rate will be obtained by calculating the number of OFP action headers received in all PACKET_OUT messages during the test duration.

### 10.4.4 Path Provisioning Time

Procedure:

```
       Test           Test               OpenFlow             OpenFlow
       Port 1        Port 2            Switches            Controller
         |             |                      |                   |
         |             |G-ARP (D1)            |                   |
         |             |---------------------->|                   |
         |             |                      |                   |
         |             |                      |PACKET_IN(D1)      |
         |             |                      |------------------>|
         |             |                      |                   |
         |Traffic (S1,D1)                     |                   |
   (Tsf1)|----------------------------------->|                   |
         |             |                      |                   |
         |             |                      |                   |
         |             |                      |                   |
         |             |                      |PACKET_IN(S1,D1)   |
         |             |                      |------------------>|
         |             |                      |                   |
         |             |                      |  FLOW_MOD(D1)     |
         |             |                      |<------------------|
         |             |                      |                   |
         |             |Traffic (S1,D1)       |                   |
         |       (Tdf1)|<---------------------|                   |
         |             |                      |                   |
```

Legend:

G-ARP: Gratuitous ARP message.<br>
Tsf1: Time of first frame sent from TP1<br>
Tdf1: Time of first frame received from TP2

Discussion:

The procedure defined above provides test steps to obtain Path Provisioning Time in reactive flow setup mode. The Path Provisioning Time can be obtained by finding the time difference between the transmit and receive time of the traffic (Tsf1-Tdf1).

### 10.4.5 Path Provisioning Rate

   Procedure:

```
       Test           Test                 OpenFlow             OpenFlow
       Port 1        Port 2              Switches            Controller
         |             |                    |                     |
         |             |                    |                     |
         |             |                    |                     |
         |             |G-ARP (D1..Dn)      |                     |
         |             |--------------------|                     |
         |             |                    |                     |
         |             |                    |PACKET_IN(D1)        |
         |             |                    |-------------------->|
         |             |                    |                     |
         |Traffic (S1..Sn,D1..Dn)           |                     |
         |---------------------------------->|                     |
         |             |                    |                     |
         |             |                    |PACKET_IN(S1..Sn,D1) |
         |             |                    |-------------------->|
         |             |                    |                     |
         |             |                    |        FLOW_MOD(S1) |
         |             |                    |<--------------------|
         |             |                    |                     |
         |             |                    |        FLOW_MOD(D1) |
         |             |                    |<--------------------|
         |             |                    |                     |
         |             |                    |        FLOW_MOD(S2) |
         |             |                    |<--------------------|
         |             |                    |                     |
         |             |                    |        FLOW_MOD(D2) |
         |             |                    |<--------------------|
         |             |                    |             .       |
         |             |                    |             .       |
         |             |                    |                     |
         |             |                    |        FLOW_MOD(Sn) |
         |             |                    |<--------------------|
         |             |                    |                     |
         |             |                    |        FLOW_MOD(Dn) |
         |             |                    |<--------------------|
         |             |                    |                     |
         |             | Traffic (S1..Sn,   |                     |
         |             |             D1..Dn)|                     |
         |             |<-------------------|                     |
         |             |                    |                     |
         |             |                    |                     |
```

Legend:

G-ARP: Gratuitous ARP<br>
D1..Dn: Destination Endpoint 1, Destination Endpoint 2 .... Destination Endpoint n<br>
S1..Sn: Source Endpoint 1, Source Endpoint 2 .., Source Endpoint n

Discussion:

The procedure defined above provides test steps to obtain Path Provisioning Rate in reactive flow setup mode. The Path Provisioning Rate can be obtained by finding the total number of frames received at TP2 after the test duration.

### 10.4.6 Network Topology Change Detection Time

Procedure:

```
      OpenFlow                    OpenFlow                     Tester
      Switches                   Controller
         |                            |                           |
         |                            |     <Bring down a link in |
         |                            |                 switch S1>|
         |                            |                           |
      T0 |PORT_STATUS with link down  |                           |
         | from S1                    |                           |
         |--------------------------->|                           |
         |                            |                           |
         |First PACKET_OUT with LLDP  |                           |
         |to OF Switch                |                           |
      T1 |<---------------------------|                           |
         |                            |                           |
         |                            |       <Record time of 1st |
         |                            |   PACKET_OUT with LLDP T1>|
```

Discussion:

The Network Topology Change Detection Time can be obtained by finding the difference between the time the OpenFlow switch S1 sends the PORT_STATUS message (T0) and the time that the OpenFlow controller sends the first topology re-discovery message (T1) to OpenFlow switches.

## 10.5 Scalability

### 10.5.1 Control Sessions Capacity

Procedure:

```
      OpenFlow                                OpenFlow
      Switches                               Controller
         |                                       |
         |    OFPT_HELLO Exchange for Switch 1   |
         |<------------------------------------->|
         |                                       |
         |    OFPT_HELLO Exchange for Switch 2   |
         |<------------------------------------->|
         |                  .                    |
         |                  .                    |
         |                  .                    |
         |    OFPT_HELLO Exchange for Switch n   |
         |X<----------------------------------->X|
         |                                       |
```

Discussion:
The value of Switch n-1 will provide Control Sessions Capacity.

### 10.5.2 Network Discovery Size

Procedure:

```
      OpenFlow                    OpenFlow                     Tester
      Switches                   Controller
         |                            |                            |
         |                            |     <Deploy network with  |
         |                            |given no. of OF switches N>|
         |                            |                            |
         |    OFPT_HELLO Exchange     |                            |
         |<-------------------------->|                            |
         |                            |                            |
         |    PACKET_OUT with LLDP    |                            |
         |      to all switches       |                            |
         |<---------------------------|                            |
         |                            |                            |
         |         PACKET_IN with LLDP|                            |
         |          rcvd from switch-1|                            |
         |-------------------------->|                            |
         |                            |                            |
```

```
         |         PACKET_IN with LLDP|                           |
         |          rcvd from switch-2|                           |
         |--------------------------->|                           |
         |            .               |                           |
         |            .               |                           |
         |                            |                           |
         |         PACKET_IN with LLDP|                           |
         |          rcvd from switch-n|                           |
         |--------------------------->|                           |
         |                            |                           |
         |                            |      <Wait for the expiry |
         |                            |     of Test Duration (Td)>|
         |                            |                           |
         |                            |   Query the controller for|
         |                            |   discovered n/w topo.(N1)|
         |                            |<--------------------------|
         |                            |                           |
         |                            |   <If N1==N repeat Step 1 |
         |                            |with N+1 nodes until N1<N >|
         |                            |                           |
         |                            |   <If N1<N repeat Step 1  |
         |                            | with N=N1 nodes once and  |
         |                            | exit>                     |
         |                            |                           |
```

Legend:
   n/w topo: Network Topology
   OF: OpenFlow

Discussion:
   The value of N1 provides the Network Discovery Size value. The test duration can be set to the stipulated time within which the user expects the controller to complete the discovery process.

### 10.5.3 Forwarding Table Capacity

Procedure:

```
       Test                OpenFlow             OpenFlow         Tester
       Port 1              Switches            Controller
         |                  |                   |                 |
         |                  |                   |                 |
         |G-ARP (H1..Hn)    |                   |                 |
  Step 1 |----------------->|                   |                 |
         |                  |                   |                 |
         |                  |PACKET_IN(D1..Dn)  |                 |
         |                  |------------------>|                 |
         |                  |                   |                 |
```

```
  Step 2 |                  |                   |<Wait for 5 secs>|
         |                  |                   |                 |
         |                  |                   |  <Query for FWD |
         |                  |                   |          entry> |(F1)
         |                  |                   |                 |
         |                  |                   |<Wait for 5 secs>|
         |                  |                   |                 |
         |                  |                   |  <Query for FWD |
         |                  |                   |          entry> |(F2)
         |                  |                   |                 |
         |                  |                   |<Wait for 5 secs>|
         |                  |                   |                 |
         |                  |                   |  <Query for FWD |
         |                  |                   |          entry> |(F3)
         |                  |                   |                 |
         |                  |                   | <Repeat Step 2  |
         |                  |                   |until F1==F2==F3>|
         |                  |                   |                 |
```

Legend:

G-ARP: Gratuitous ARP
H1..Hn: Host 1 .. Host n
FWD: Forwarding Table

Discussion:

Query the controller forwarding table entries for multiple times until the three consecutive queries return the same value. The last value retrieved from the controller will provide the Forwarding Table Capacity value. The query interval is user configurable. The 5 seconds shown in this example is for representational purpose.

## 10.6 Security

### 10.6.1 Exception Handling

Procedure:

```
     Test        Test         OpenFlow          OpenFlow          Tester
     Port 1     Port 2       Switches          Controller
       |          |                   |                 |            |
       |          |G-ARP (D1..Dn)     |                 |            |
       |          |------------------>|                 |            |
       |          |                   |                 |            |
       |          |                   |PACKET_IN(D1..Dn)|            |
       |          |                   |---------------->|            |
       |          |                   |                 |            |
 Step 1|Traffic (S1..Sn,D1..Dn)       |                 |            |
       |----------------------------->|                 |            |
       |          |                   |                 |            |
       |          |                   |PACKET_IN(S1..Sa,|             |
       |          |                   |          D1..Da)|             |
       |          |                   |---------------->|             |
       |          |                   |                 |             |
       |          |                   |PACKET_IN(Sa+1.. |             |
       |          |                   |.Sn,Da+1..Dn)    |             |
       |          |                   |(1% incorrect OFP|             |
       |          |                   |    Match header)|             |
       |          |                   |---------------->|             |
       |          |                   |                 |             |
       |          |                   | FLOW_MOD(D1..Dn)|             |
       |          |                   |<----------------|             |
       |          |                   |                 |             |
       |          |                   | FLOW_MOD(S1..Sa)|             |
       |          |                   |      OFP headers|             |
       |          |                   |<----------------|             |
       |          |                   |                 |             |
       |          |Traffic (S1..Sa,   |                 |             |
       |          |            D1..Da)|                 |             |
       |          |<------------------|                 |             |
       |          |                   |                 |             |
       |          |                   |                 |   <Wait for |
       |          |                   |                 |       Test  |
       |          |                   |                 |    Duration>|
       |          |                   |                 |             |
       |          |                   |                 |   <Record Rx|
       |          |                   |                 |    frames at|
       |          |                   |                 |   TP2 (Rn1)>|
       |          |                   |                 |             |
       |          |                   |                 |     <Repeat |
       |          |                   |                 |  Step1 with |
       |          |                   |                 | 2% incorrect|
       |          |                   |                 |  PACKET_INs>|
       |          |                   |                 |             |
       |          |                   |                 |   <Record Rx|
       |          |                   |                 |    frames at|
       |          |                   |                 |   TP2 (Rn2)>|
       |          |                   |                 |             |

   Legend:
      G-ARP: Gratuitous ARP
      PACKET_IN(Sa+1..Sn,Da+1..Dn): OpenFlow PACKET_IN with wrong
            version number
      Rn1: Total number of frames received at Test Port 2 with
           1% incorrect frames
      Rn2: Total number of frames received at Test Port 2 with
           2% incorrect frames
```

Discussion:

The traffic rate sent towards OpenFlow switch from Test Port 1 should be 1% higher than the Path Programming Rate. Rn1 will provide the Path Provisioning Rate of controller at 1% of incorrect frames handling and Rn2 will provide the Path Provisioning Rate of controller at 2% of incorrect frames handling.

The procedure defined above provides test steps to determine the effect of handling error packets on Path Programming Rate. Same procedure can be adopted to determine the effects on other performance tests listed in this benchmarking tests.

### 10.6.2 Denial of Service Handling

Procedure:

```
     Test          Test         OpenFlow          OpenFlow       Tester
     Port 1        Port 2       Switches          Controller
       |          |                   |                 |            |
       |          |G-ARP (D1..Dn)     |                 |            |
       |          |------------------>|                 |            |
       |          |                   |                 |            |
       |          |                   |PACKET_IN(D1..Dn)|            |
       |          |                   |---------------->|            |
       |          |                   |                 |            |
       |Traffic (S1..Sn,D1..Dn)       |                 |            |
       |----------------------------->|                 |            |
       |          |                   |                 |            |
       |          |                   |PACKET_IN(S1..Sn,|            |
       |          |                   |          D1..Dn)|            |
       |          |                   |---------------->|            |
       |          |                   |                 |            |
       |          |                   |TCP SYN Attack   |            |
       |          |                   |from a switch    |            |
       |          |                   |---------------->|            |
       |          |                   |                 |            |
       |          |                   |FLOW_MOD(D1..Dn) |            |
       |          |                   |<----------------|            |
       |          |                   |                 |            |
       |          |                   | FLOW_MOD(S1..Sn)|            |
       |          |                   |      OFP headers|            |
       |          |                   |<----------------|            |
       |          |                   |                 |            |
```

```
       |          |Traffic (S1..Sn,   |                 |            |
       |          |            D1..Dn)|                 |            |
       |          |<------------------|                 |            |
       |          |                   |                 |            |
       |          |                   |                 |  <Wait for |
       |          |                   |                 |      Test  |
       |          |                   |                 |   Duration>|
       |          |                   |                 |            |
       |          |                   |                 |  <Record Rx|
       |          |                   |                 |   frames at|
       |          |                   |                 |  TP2 (Rn1)>|
       |          |                   |                 |            |
```

Legend:
G-ARP: Gratuitous ARP

Discussion:
TCP SYN attack should be launched from one of the emulated/simulated OpenFlow Switch. Rn1 provides the Path Programming Rate of controller uponhandling denial of service attack.

The procedure defined above provides test steps to determine the effect of handling denial of service on Path Programming Rate. Same procedure can be adopted to determine the effects on other performance tests listed in this benchmarking tests.

### 10.7 Reliability

#### 10.7.1 Controller Failover Time

Procedure:

```
     Test          Test         OpenFlow          OpenFlow       Tester
     Port 1        Port 2       Switches          Controller
       |             |             |                 |               |
       |             |G-ARP (D1)   |                 |               |
       |             |------------>|                 |               |
       |             |             |                 |               |
       |             |             |PACKET_IN(D1)    |               |
       |             |             |---------------->|               |
       |             |             |                 |               |
 Step 1|Traffic (S1..Sn,D1)        |                 |               |
       |-------------------------->|                 |               |
       |             |             |                 |               |
```

```
       |             |             |                 |               |
       |             |             |PACKET_IN(S1,D1) |               |
       |             |             |---------------->|               |
       |             |             |                 |               |
       |             |             |FLOW_MOD(D1)     |               |
       |             |             |<----------------|               |
       |             |             |FLOW_MOD(S1)     |               |
       |             |             |<----------------|               |
       |             |             |                 |               |
       |             |Traffic (S1,D1)|               |               |
       |             |<------------|                 |               |
       |             |             |                 |               |
       |             |             |PACKET_IN(S2,D1) |               |
       |             |             |---------------->|               |
       |             |             |                 |               |
       |             |             |FLOW_MOD(S2)     |               |
       |             |             |<----------------|               |
       |             |             |                 |               |
       |             |             |PACKET_IN(Sn-1,D1)|              |
       |             |             |---------------->|               |
       |             |             |                 |               |
       |             |             |PACKET_IN(Sn,D1) |               |
       |             |             |---------------->|               |
       |             |             |       .         |               |
       |             |             |       .         |<Bring down the|
       |             |             |       .         |active control-|
       |             |             |                 |       ler>    |
       |             |             |  FLOW_MOD(Sn-1) |               |
       |             |             |    <-X----------|               |
       |             |             |                 |               |
       |             |             |FLOW_MOD(Sn)     |               |
       |             |             |<----------------|               |
       |             |             |                 |               |
       |             |Traffic (Sn,D1)|               |               |
       |             |<------------|                 |               |
       |             |             |                 |               |
       |             |             |                 |<Stop the test |
       |             |             |                 |after recv.    |
       |             |             |                 |traffic upon   |
       |             |             |                 | failure>      |
```

Legend:

G-ARP: Gratuitous ARP.

Discussion:

The time difference between the last valid frame received before the traffic loss and the first frame received after the traffic loss will provide the controller failover time.

If there is no frame loss during controller failover time, the controller failover time can be deemed negligible.

### 10.7.2 Network Re-Provisioning Time

Procedure:

```
     Test          Test         OpenFlow          OpenFlow        Tester
     Port 1        Port 2       Switches          Controller
       |             |               |                 |               |
       |             |G-ARP (D1)     |                 |               |
       |             |-------------->|                 |               |
       |             |               |                 |               |
       |             |               |PACKET_IN(D1)    |               |
       |             |               |---------------->|               |
       |              G-ARP (S1)     |                 |               |
       |---------------------------->|                 |               |
       |             |               |                 |               |
       |             |               |PACKET_IN(S1)    |               |
       |             |               |---------------->|               |
       |             |               |                 |               |
       |Traffic (S1,D1,Seq.no (1..n))|                 |               |
       |---------------------------->|                 |               |
       |             |               |                 |               |
       |             |               |PACKET_IN(S1,D1) |               |
       |             |               |---------------->|               |
       |             |               |                 |               |
       |             |Traffic (D1,S1,|                 |               |
       |             | Seq.no (1..n))|                 |               |
       |             |-------------->|                 |               |
       |             |               |                 |               |
       |             |               |PACKET_IN(D1,S1) |               |
       |             |               |---------------->|               |
       |             |               |                 |               |
       |             |               |FLOW_MOD(D1)     |               |
       |             |               |<----------------|               |
       |             |               |                 |               |
       |             |               |FLOW_MOD(S1)     |               |
       |             |               |<----------------|               |
       |             |               |                 |               |
       |             |Traffic (S1,D1,|                 |               |
       |             |     Seq.no(1))|                 |               |
       |             |<--------------|                 |               |
       |             |               |                 |               |
       |             |Traffic (S1,D1,|                 |               |
       |             |     Seq.no(2))|                 |               |
       |             |<--------------|                 |               |
       |             |               |                 |               |
```

```
       |             |               |                 |              |
       |    Traffic (D1,S1,Seq.no(1))|                 |              |
       |<----------------------------|                 |              |
       |             |               |                 |              |
       |    Traffic (D1,S1,Seq.no(2))|                 |              |
       |<----------------------------|                 |              |
       |             |               |                 |              |
       |    Traffic (D1,S1,Seq.no(x))|                 |              |
       |<----------------------------|                 |              |
       |             |               |                 |              |
       |             |Traffic (S1,D1,|                 |              |
       |             |     Seq.no(x))|                 |              |
       |             |<--------------|                 |              |
       |             |               |                 |              |
       |             |               |                 |              |
       |             |               |                 |  <Bring down |
       |             |               |                 | the switch in|
       |             |               |                 |active traffic|
       |             |               |                 |       path>  |
       |             |               |                 |              |
       |             |               |PORT_STATUS(Sa)  |              |
       |             |               |---------------->|              |
       |             |               |                 |              |
       |             |Traffic (S1,D1,|                 |              |
       |             |   Seq.no(n-1))|                 |              |
       |             |  X<-----------|                 |              |
       |             |               |                 |              |
       |  Traffic (D1,S1,Seq.no(n-1))|                 |              |
       |    X------------------------|                 |              |
       |             |               |                 |              |
       |             |               |FLOW_MOD(D1)     |              |
       |             |               |<----------------|              |
       |             |               |                 |              |
       |             |               |FLOW_MOD(S1)     |              |
       |             |               |<----------------|              |
       |             |               |                 |              |
       |    Traffic (D1,S1,Seq.no(n))|                 |              |
       |<----------------------------|                 |              |
       |             |               |                 |              |
       |             |Traffic (S1,D1,|                 |              |
       |             |     Seq.no(n))|                 |              |
       |             |<--------------|                 |              |
       |             |               |                 |              |
       |             |               |                 |<Stop the test|
       |             |               |                 |  after recv. |
       |             |               |                 |  traffic upon|
       |             |               |                 |   failover>  |
```

Legend:

G-ARP: Gratuitous ARP message.<br>
Seq.no: Sequence number.<br>
Sa: Neighbour switch of the switch that was brought down.

Discussion:

The time difference between the last valid frame received before the traffic loss (Packet number with sequence number x) and the first frame received after the traffic loss (packet with sequence number n) will provide the network path re-provisioning time.

Note that the test is valid only when the controller provisions the alternate path upon network failure.

## 11. Acknowledgements

The authors would like to acknowledge Sandeep Gangadharan (HP) for the significant contributions to the current and earlier versions of this document. The authors would like to thank the following individuals for providing their valuable comments to the earlier versions of this document: Al Morton (AT&T), M. Georgescu (NAIST), Andrew McGregor (Google), Scott Bradner (Harvard University), Jay Karthik (Cisco), Ramakrishnan (Brocade).

## 12. Authors' Addresses

Bhuvaneswaran Vengainathan
Veryx Technologies Inc.
1 International Plaza, Suite 550
Philadelphia
PA 19113

Email: bhuvaneswaran.vengainathan@veryxtech.com

Anton Basil
Veryx Technologies Inc.
1 International Plaza, Suite 550
Philadelphia
PA 19113

Email: anton.basil@veryxtech.com

Mark Tassinari
Hewlett-Packard,
8000 Foothills Blvd,
Roseville, CA 95747

Email: mark.tassinari@hp.com

Vishwas Manral
Ionos Corp,
4100 Moorpark Ave,
San Jose, CA

Email: vishwas@ionosnetworks.com

Sarah Banks
VSS Monitoring

Email: sbanks@encrypted.net