

Internet-Draft
Vengainathan
Network Working Group
Basil
Intended Status: Informational
Technologies
Expires: March 2015
Manral
Corp
Tassinari
Packard
2014

Bhuvaneshwaran
Anton
Veryx
Vishwas
Ionos
Mark
Hewlett-
September 26,

**Benchmarking Methodology for SDN Controller Performance
draft-bhuvan-bmwg-of-controller-benchmarking-01**

Abstract

This document defines the metrics and methodologies for measuring performance of SDN controllers. SDN controllers have been implemented with many varying designs, in order to achieve their intended network functionality. Hence, in this document the authors take the approach of considering an SDN controller as a black box, defining the metrics 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

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 26, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Bhuvan, et al.

Expires March 26, 2015

[Page 1]

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1.](#) Introduction [3](#)
- [2.](#) Terminology [3](#)
- [3.](#) Scope [4](#)
- [4.](#) Test Setup [5](#)
 - [4.1](#) SDN Network - Controller working in Standalone Mode [5](#)
 - [4.2](#) SDN Network - Controller working in Cluster Mode [5](#)
 - [4.3](#) SDN Network with TE - Controller working in Standalone Mode. [6](#)
 - [4.4](#) SDN Network with TE - Controller working in Cluster Mode . . [6](#)
 - [4.5](#) SDN Node with TE - Controller working in Standalone Mode . . [7](#)
 - [4.6](#) SDN Node with TE - Controller working in Cluster Mode . . . [8](#)
- [5.](#) Test Considerations [8](#)
 - [5.1](#) Network Topology [8](#)
 - [5.2](#) Test Traffic [8](#)
 - [5.3](#) Connection Setup [9](#)
 - [5.4](#) Measurement Accuracy [9](#)
 - [5.5](#) Real World Scenario [9](#)
- [6.](#) Test Reporting [9](#)
- [7.](#) Benchmarking Tests [10](#)
 - [7.1](#) Performance [10](#)
 - [7.1.1](#) Network Topology Discovery Time [10](#)
 - [7.1.2](#) Synchronous Message Processing Time [12](#)
 - [7.1.3](#) Synchronous Message Processing Rate [13](#)
 - [7.1.4](#) Path Provisioning Time [15](#)
 - [7.1.5](#) Path Provisioning Rate [17](#)
 - [7.1.6](#) Network Topology Change Detection Time [19](#)
 - [7.2](#) Scalability [21](#)
 - [7.2.1](#) Network Discovery Size [21](#)
 - [7.2.2](#) Flow Scalable Limit [22](#)
 - [7.3](#) Security [23](#)
 - [7.3.1](#) Exception Handling [23](#)
 - [7.3.2](#) Denial of Service Handling [24](#)
 - [7.4](#) Reliability [25](#)
 - [7.4.1](#) Controller Failover Time [25](#)
 - [7.4.2](#) Network Re-Provisioning Time [26](#)
- [8.](#) Test Coverage [28](#)

[9.](#) References [28](#)
[9.1](#) Normative References [28](#)
[9.2](#) Informative References [29](#)
[10.](#) IANA Considerations [29](#)
[11.](#) Security Considerations [29](#)
[12.](#) Acknowledgements [29](#)
[13.](#) Authors' Addresses [30](#)

[1.](#) Introduction

This document provides generic metrics and methodologies for benchmarking SDN controller performance. An SDN controller may support many northbound and southbound protocols, implement wide range of applications and work as standalone 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 various controller designs 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.](#) Terminology

SDN Node:

An SDN node is a physical or virtual entity that forwards data in a software defined environment.

Flow:

A flow is a traffic stream having same source and destination address. The address could be MAC or IP or combination of both.

Learning Rate:

The rate at which the controller learns the new source addresses from the received traffic without dropping.

Controller Forwarding Table:

A controller forwarding table contains flow records for the flows configured in the data path.

Northbound Interface:

Northbound interface is the application programming interface provided by the SDN controller for communication with SDN services and applications.

Southbound Interface:

Southbound interface is the application programming interface provided by the SDN controller for communication with the SDN nodes.

Proactive Flow Provisioning:

Proactive flow provisioning is the pre-provisioning of flow entries into the controller's forwarding table through controller's northbound interface or management interface.

Reactive Flow Provisioning:

Reactive flow provisioning is the dynamic provisioning of flow entries into the controller's forwarding table based on traffic forwarded by the SDN nodes through controller's southbound interface.

Path:

A path is the route taken by a flow while traversing from a source node to destination node.

Standalone Mode:

Single controller handling all control plane functionalities.

Cluster/Redundancy Mode:

Group of controllers handling all control plane functionalities .

Synchronous Message:

Any message from the SDN node that triggers a response message from the controller e.g., Keepalive request and response message, flow setup request and response message etc.,

3. Scope

This document defines a number of tests to measure the networking aspects of SDN controllers. These tests are recommended for execution in lab environments rather than in real time deployments.

4. 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.

4.1 SDN Network - Controller working in Standalone Mode

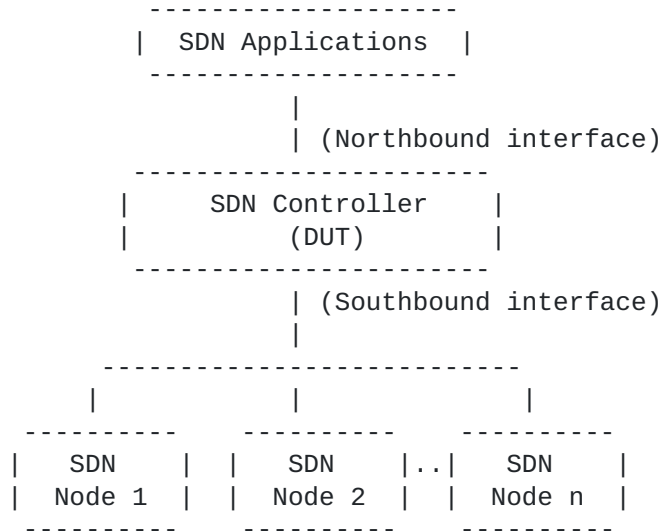


Figure 1

4.2 SDN Network - Controller working in Cluster Mode

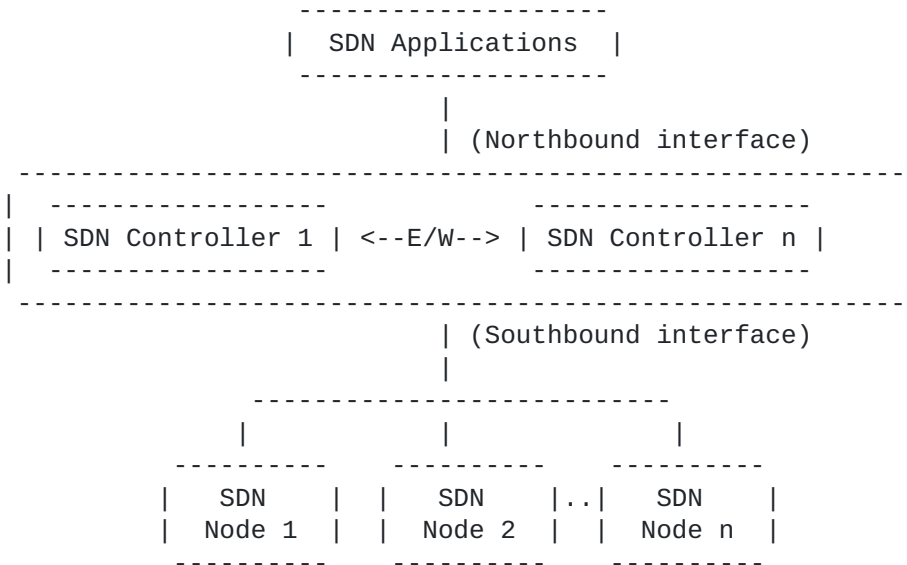


Figure 2

4.3 SDN Network with Traffic Endpoints (TE) - Controller working in Standalone Mode

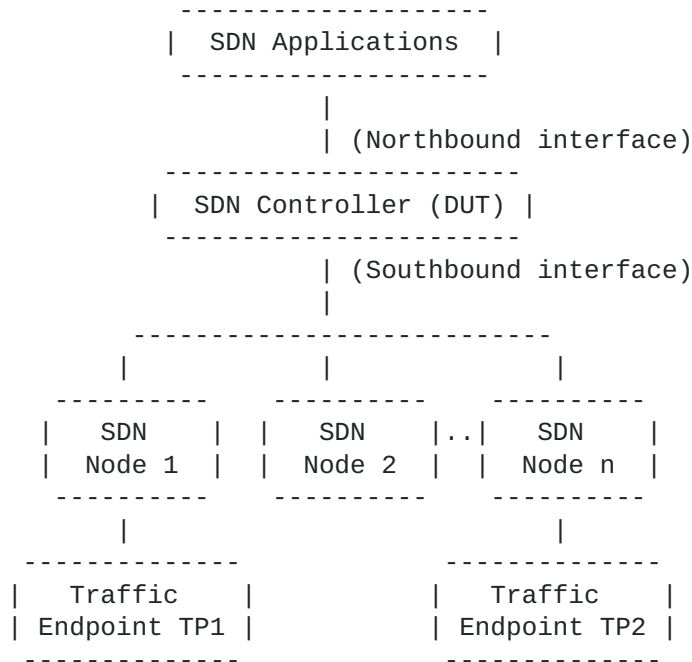
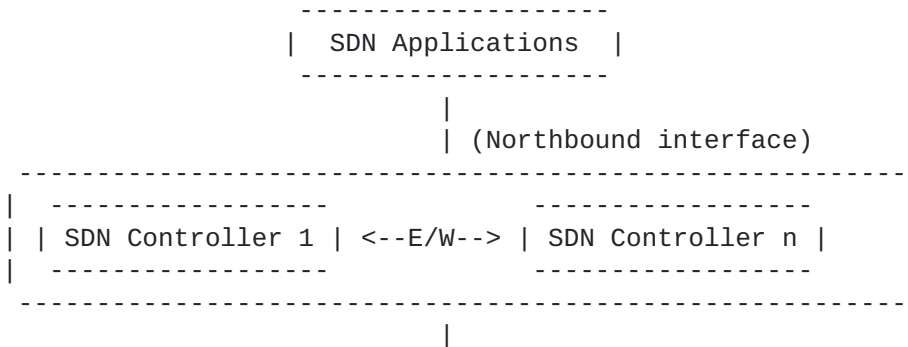


Figure 3

4.4 SDN Network with Traffic Endpoints (TE) - Controller working in Cluster Mode



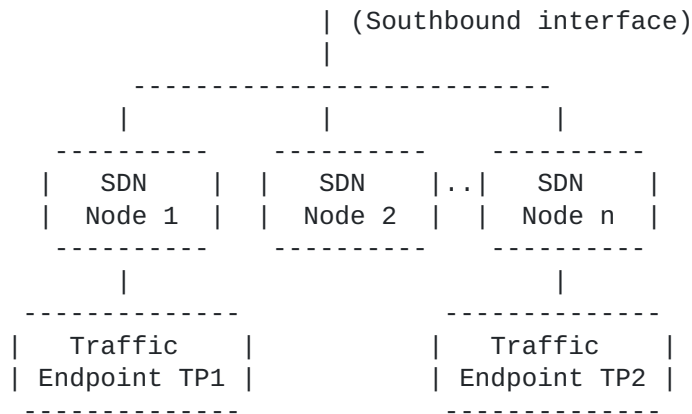


Figure 4

4.5 SDN Node with Traffic Endpoints (TE) - Controller working in Standalone Mode

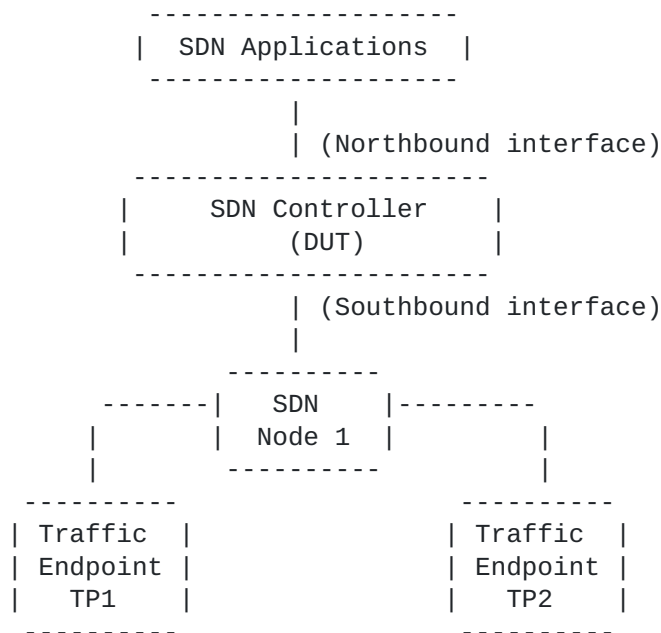


Figure 5

4.6 SDN Node with Traffic Endpoints (TE) - Controller working in Cluster Mode

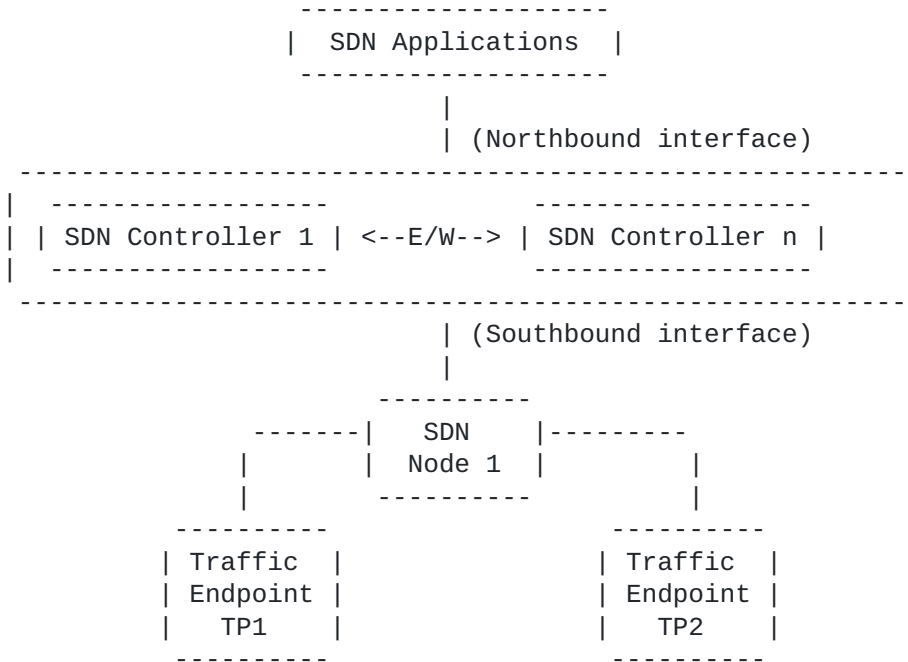


Figure 6

5. Test Considerations

5.1 Network Topology

The network SHOULD be deployed with SDN nodes interconnected in either fully meshed, tree or linear topology. Care should be taken to make sure that the loop prevention mechanism is enabled either in the SDN controller or in the network. To get complete performance characterization of 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.

5.2 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 that at least five different frame sizes and traffic types be used, depending on the intended network deployment.

5.3 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 the applicable connection setup methods.

1. Unencrypted connection with SDN nodes, running same protocol version.
2. Unencrypted connection with SDN nodes, running different (previous) protocol versions.
3. Encrypted connection with SDN nodes, running same protocol version
4. Encrypted connection with SDN nodes, running different (previous) protocol versions.

5.4 Measurement Accuracy

The measurement accuracy depends on 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.

5.5 Real World Scenario

Benchmarking tests discussed in the document are to be performed on a "black-box" basis, relying solely on measurements observable external to the controller. The network deployed and the test parameters should be identical to the deployment scenario to obtain value added measures.

6. Test Reporting

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

1. Controller name and version
2. Northbound protocols and version
3. Southbound protocols and version
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

7. Benchmarking Tests

7.1 Performance

7.1.1 Network Topology Discovery Time

Objective:

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

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.

Test Interval (To)- Defines the maximum time for the test to complete, expressed in milliseconds.

Test Setup:

The test can use one of the test setup described in [section 4.1](#) and 4.2 of this document.

Prerequisite:

1. The controller should support network discovery.
2. Tester should be able to retrieve the discovered topology information either through controller's management interface or northbound interface.

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 exchange between the controller and the network node (Tm1).
5. Query the controller continuously for the discovered network topology information 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 expiry of test interval (To).

7. Record the time last discovery message exchange between the controller and the network node (Tmn) when the test completed successfully.

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

Measurement:

Topology Discovery Time $Tr1 = Tmn - Tm1$.

$$\text{Average Topology Discovery Time} = \frac{Tr1 + Tr2 + Tr3 \dots Trn}{\text{Total Test Iterations}}$$

Note:

1. To increase the certainty of 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 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 SHOULD be reported in the format of a table, with a row for each 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.

7.1.2 Synchronous Message Processing Time

Objective:

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

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.

Test Duration (Td) - Defines the duration of test iteration, expressed in seconds. The recommended value is 5 seconds.

Test Setup:

The test can use one of the test setup described in [section 4.1](#) and 4.2 of this document.

Prerequisite:

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

Procedure:

1. Generate a synchronous message from every connected nodes one at a time and wait for the response before generating the next message.
2. Record total number of messages sent to the controller by all nodes (Ntx) and the responses received from the controller (Nrx) within the test duration (Td).

Measurement:

$$\text{Synchronous Message Processing Time } Tr1 = \frac{Td}{Nrx}$$

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

Note:

1. The above test measures the controller's message processing time at lower traffic rate. To measure the controller's message processing time at full connection rate, apply the same measurement equation with the Td and Nrx values obtained from Synchronous Message Processing Rate test (defined in [Section 7.1.3](#)).
2. To increase the certainty of 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 synchronous 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 Synchronous Message Processing Time results SHOULD be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Synchronous Message Processing Time.

The report should capture the following information in addition to the configuration parameters captured in [section 6](#).
- Offered rate (Ntx)

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 Synchronous 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 Synchronous Message Processing Time.

7.1.3 Synchronous Message Processing Rate

Objective:

To measure the maximum number of synchronous 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.

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology.

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.

Test Duration (Td) - Defines the duration of test iteration, expressed in seconds. The recommended value is 5 seconds.

Test Setup:

The test can use one of the test setup described in [section 4.1](#) and 4.2 of this document.

Prerequisite:

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

Procedure:

1. Generate synchronous messages from all the connected nodes at the full connection capacity for the Test Duration (Td).
2. Record total number of messages sent to the controller by all nodes (Ntx) and the responses received from the controller (Nrx) within the test duration (Td).

Measurement:

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

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

Note:

1. To increase the certainty of 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 synchronous 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 Synchronous Message Processing Rate results SHOULD be reported in the format of a table with a row for each iteration. The last row of the table indicates the average Synchronous Message Processing Rate.

The report should capture the following information in addition to the configuration parameters captured in [section 6](#).
- Offered rate (Ntx)

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 Synchronous 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 Synchronous Message Processing Rate.

[7.1.4](#) Path Provisioning Time

Objective:

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

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology
Number of data path nodes (Ndp) - Defines the number of nodes present in the path between source and destination node.

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.
Test Interval (To) - Defines the maximum time for the test to complete, expressed in milliseconds.

Test Setup:

The test can use one of the test setups described in [section 4.3](#) and 4.4 of this document.

Prerequisite:

1. The controller should 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 learnt about the location of source/destination endpoint for which the path has to be provisioned. This can be achieved through dynamic learning or static provisioning.
4. The SDN Node should send all new flows to the controller when it receives.

Procedure:

Reactive Path Provisioning:

1. Send traffic with source as source endpoint address and destination as destination endpoint address from TP1.
2. Record the time for the first frame sent to the source SDN node (Tsf1).
3. Wait for the arrival of first frame from the destination node or the expiry of test interval (To).
4. Record the time when the first frame received from the destination SDN node (Tdf1).

Proactive Path Provisioning:

1. Send traffic with source as source endpoint address and destination as destination endpoint address from TP1.
2. Install the flow with the learnt source and destination address through controller's northbound or management interface.
3. Record the time when a successful response for the flow installation is received (Tp) from the controller.
4. Wait for the arrival of first frame from the destination node or the expiry of test interval (To).
5. Record the time when the first frame received from the destination node (Tdf1).

Measurement:

Reactive Path Provisioning:

$$\text{Flow Provisioning Time } Tr1 = Tdf1 - Tsf1.$$

Proactive Path Provisioning:

$$\text{Path Provisioning Time } Tr1 = Tdf1 - Tp.$$

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

Note:

1. To increase the certainty of 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 SHOULD 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 6](#).

- 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.

7.1.5 Path Provisioning Rate

Objective:

To measure the maximum number of paths a controller can setup between sources and destination node within the test duration, expressed in paths per second.

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology.

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.

Test Duration (Td)- Defines the duration of test iteration, expressed in seconds. The recommended value is 5 seconds.

Test Setup:

The test can use one of the test setup described in [section 4.3](#) and 4.4 of this document.

Prerequisite:

1. The controller should 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 learnt about the location of source/destination endpoints for which the paths have to be provisioned. This can be achieved through dynamic learning or static provisioning.
4. The SDN Node should send all new flows to the controller when it receives.

Procedure:

Reactive Path Provisioning:

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

Proactive Path Provisioning:

1. Send traffic continuously with unique source and destination addresses from the source node.
2. Install flows with the learnt source and destination addresses 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:

Proactive/Reactive Path Provisioning:

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

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

Note:

1. To increase the certainty of 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 SHOULD 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 6](#).

- 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.

7.1.6 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.

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the number of nodes present in the defined network topology

Test setup parameters:

Test Iterations (Tr) - Defines the number of times the test needs to be repeated. The recommended value is 3.

Test Interval (To) - Defines the maximum time for the test to complete, expressed in milliseconds. Test not completed within

this

time interval is considered as incomplete.

Test Setup:

The test can use one of the test setup described in [section 4.1](#) and 4.2 of this document.

Prerequisite:

1. The controller should have discovered the network topology information for the deployed network topology.
2. The periodic network discovery operation should be configured to twice the Test Interval (To) 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 $Tr_1 = Tcd - Tcn$.

$$\text{Average Network Topology Change Detection Time} = \frac{Tr_1 + Tr_2 + Tr_3 \dots Tr_n}{\text{Total Test Iterations}}$$

Note:

1. To increase the certainty of 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 SHOULD 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.

7.2 Scalability

7.2.1 Network Discovery Size

Objective:

To measure the network size (number of nodes) that a controller can discover within a stipulated time.

Setup Parameters:

The following parameters MUST be defined:

Network setup parameters:

Number of nodes (N) - Defines the initial number of nodes present in the defined network topology

Test setup parameters:

Network Discovery Time (Tnd) - Defines the stipulated time acceptable by the user, expressed in seconds.

Test Setup:

The test can use one of the test setup described in [section 4.1](#) and 4.2 of this document.

Prerequisite:

1. The controller should 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 after the expiry of Network Discovery Time (Tnd).
3. Increase the number of nodes by 1 when the comparison is successful and repeat the test.
4. Decrease the number of nodes by 1 when the comparison fails and repeat the test.
5. Continue the test until the comparison of step 4 is successful.
6. Record the number of nodes for the last iteration (Ns) where the topology comparison was successful.

Measurement:

Network Discovery Size = N_s .

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 SHOULD be reported in addition to the configuration parameters captured in [section 6](#).

7.2.2 Flow Scalable Limit

Objective:

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

Setup Parameters:

The following parameters MUST be defined:

Test Setup:

The test can use one of the test setups described in [section 4.5](#) and 4.6 of this document.

Prerequisite:

1. The controller Forwarding table should be empty.
2. Flow Idle time should be set to higher or infinite value.
3. The controller should 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 Path Provisioning:

1. Send bi-directional traffic continuously with unique source and/or destination addresses from test ports TP1 and TP2 at the learning 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 (N_{rp}).

Proactive Path Provisioning:

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 path provisioning 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 Path Provisioning:

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

Reactive Path Provisioning:

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

Flow Scalable Limit = Max Flow Entries.

Reporting Format:

The Flow Scalable Limit results SHOULD be tabulated with the following information in addition to the configuration parameters captured in [section 6](#).

- Provisioning Type (Proactive/Reactive)

7.3 Security

7.3.1 Exception Handling

Objective:

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

- a. Path Programming Rate
- b. Path Programming Time
- c. Network Topology Change Detection Time

Prerequisite:

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

Procedure:

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

Note:

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

Measurement:

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

Reporting Format:

The Exception Handling results SHOULD 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

7.3.2 Denial of Service Handling

Objective:

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

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

Prerequisite:

This test should 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 should be done as per the equation defined in the corresponding test's measurement section.

Reporting Format:

The DoS Attacks Handling results SHOULD 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.

7.4 Reliability

7.4.1 Controller Failover Time

Objective:

To compute the time taken to switch from one controller to another when the controllers are teamed and the active controller fails.

Setup Parameters:

The following parameters MUST be defined:

Controller setup parameters:

Number of cluster nodes (CN) - Defines the number of member nodes present in the cluster.

Redundancy Mode (RM) - Defines the controller clustering mode e.g., Active - Standby or Active - Active.

Test Setup:

The test can use the test setup described in [section 4.4](#) of this document.

Prerequisite:

1. Master controller election should 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 should send all new flows to the controller when it receives.

Procedure:

1. Send bi-directional traffic continuously with unique source and/or destination addresses from test ports TP1 and TP2 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 test duration (Td), total number of frames sent (Nsnt) on TP1 and number of frames received (Nrvd) on TP2.

Measurement:

$$\text{Controller Failover Time} = ((Td/Nrvd) - (Td/Nsnt))$$
$$\text{Packet Loss} = Nsnt - Nrvd$$

Reporting Format:

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

- Number of cluster nodes
- Redundancy mode
- Controller Failover
- Time Packet Loss

7.4.2 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.

Setup Parameters:

Same setup parameters as defined in the Path Programming Rate performance test ([Section 7.1.5](#)).

Prerequisite:

Network with the given number of nodes and intended topology (Mesh or Tree) with redundant paths should be deployed.

Procedure:

1. Perform the test procedure mentioned in Path Programming Rate test ([Section 7.1.5](#)).
2. Send bi-directional traffic continuously with unique sequence number for one particular traffic endpoint.
3. Bring down a link or switch in the traffic path.
4. Stop the test after receiving first frame after network re-convergence (timeline).
5. 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).
6. 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).

Measurement:

$$\text{Forward Direction Path Re-Provisioning Time (FDRT)} \\ = (\text{TP2-Tffr} - \text{TP2-Tlfr})$$

$$\text{Reverse Direction Path Re-Provisioning Time (RDRT)} \\ = (\text{TP1-Tffr} - \text{TP1-Tlfr})$$

Tlfr)

$$\text{Network Re-Provisioning Time} = (\text{FDRT} + \text{RDRT}) / 2$$

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 SHOULD 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

8. Test Coverage

	Performance	Scalability	Reliability
Setup	1. Network Topology Discovery 2. Path Provisioning Time 3. Path Provisioning Rate	1. Network Discovery Size	
Operational	1. Synchronous Message Processing Rate 2. Synchronous Message Processing Time	1. Flow Scalable Limit	1. Network Topology Change Detection Time 2. Exception Handling 3. Denial of Service Handling 4. Network Re-Provisioning Time
Tear Down			1. Controller Failover Time

9. References

9.1 Normative References

[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.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-05](#) (Work in progress), July 20, 2014.

9.2 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

10. IANA Considerations

This document does not have any IANA requests.

11. 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.

12. Acknowledgements

The authors would like to acknowledge the following individuals for their help and participation of the compilation of this document:
Al Morton (AT&T), Brain Castelli (Spirent), Sandeep Gangadharan(HP),
Sarah Banks (VSS Monitoring) who made significant suggestions to the
current and earlier versions of this document.

13. Authors' Addresses

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

Email: bhuvaneshwaran.vengainathan@veryxtech.com

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

Email: anton.basil@veryxtech.com

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

Email: vishwas@ionosnetworks.com

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

Email: mark.tassinari@hp.com