Internet Engineering Task Force (IETF)
Request for Comments: 8455
Category: Informational
ISSN: 2070-1721

V. Bhuvaneswaran
A. Basil
Veryx Technologies
M. Tassinari
Hewlett Packard Enterprise
V. Manral
NanoSec
S. Banks
VSS Monitoring

October 2018

Terminology for Benchmarking Software-Defined Networking (SDN) Controller Performance

Abstract

This document defines terminology for benchmarking a Software-Defined Networking (SDN) controller's control-plane performance. It extends the terminology already defined in RFC 7426 for the purpose of benchmarking SDN Controllers. The terms provided in this document help to benchmark an SDN Controller's performance independently of the controller's supported protocols and/or network services.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8455.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://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
1.1. Conventions Used in This Document 3
2. Term Definitions 4
2.1. SDN Terms 4
2.1.1. Flow 4
2.1.2. Northbound Interface 4
2.1.3. Southbound Interface 5
2.1.4. Controller Forwarding Table 5
2.1.5. Proactive Flow Provisioning Mode 5
2.1.6. Reactive Flow Provisioning Mode 6
2.1.7. Path 6
2.1.8. Standalone Mode 6
2.1.9. Cluster/Redundancy Mode 7
2.1.10. Asynchronous Message 7
2.1.11. Test Traffic Generator 7
2.1.12. Leaf-Spine Topology 8
2.2. Test Configuration/Setup Terms 8
2.2.1. Number of Network Devices 8
2.2.2. Trial Repetition 8
2.2.3. Trial Duration 9
2.2.4. Number of Cluster Nodes 9
2.3. Benchmarking Terms 9
2.3.1. Performance 9
2.3.1.1. Network Topology Discovery Time 9
2.3.1.2. Asynchronous Message Processing Time 10
2.3.1.3. Asynchronous Message Processing Rate 10
2.3.1.4. Reactive Path Provisioning Time 11
2.3.1.5. Proactive Path Provisioning Time 12
2.3.1.6. Reactive Path Provisioning Rate 12
2.3.1.7. Proactive Path Provisioning Rate 13
2.3.1.8. Network Topology Change Detection Time 13

2.3.2. Scalability 14
2.3.2.1. Control Sessions Capacity 14
2.3.2.2. Network Discovery Size 14
2.3.2.3. Forwarding Table Capacity 15
2.3.3. Security 15
2.3.3.1. Exception Handling 15
2.3.3.2. Handling Denial-of-Service Attacks 16
2.3.4. Reliability 16
2.3.4.1. Controller Failover Time 16
2.3.4.2. Network Re-provisioning Time 17
3. Test Setup 17
3.1. Test Setup - Controller Operating in Standalone Mode 18
3.2. Test Setup - Controller Operating in Cluster Mode 19
4. Test Coverage 20
5. IANA Considerations 21
6. Security Considerations 21
7. Normative References 21
Acknowledgments 22
Authors' Addresses 23

1. Introduction

Software-Defined Networking (SDN) is a networking architecture in which network control is decoupled from the underlying forwarding function and is placed in a centralized location called the SDN Controller. The SDN Controller provides an abstraction of the underlying network and offers a global view of the overall network to applications and business logic. Thus, an SDN Controller provides the flexibility to program, control, and manage network behavior dynamically through northbound and southbound interfaces. Since the network controls are logically centralized, the need to benchmark the SDN Controller's performance becomes significant. This document defines terms to benchmark various controller designs for performance, scalability, reliability, and security, independently of northbound and southbound protocols. A mechanism for benchmarking the performance of SDN Controllers is defined in the companion methodology document [RFC8456]. These two documents provide methods for measuring and evaluating the performance of various controller implementations.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Term Definitions

2.1. SDN Terms

The terms defined in this section are extensions to the terms defined in [RFC7426] ("Software-Defined Networking (SDN): Layers and Architecture Terminology"). Readers should refer to [RFC7426] before attempting to make use of this document.

2.1.1. Flow

Definition:
The definition of "flow" is the same as the definition of "microflows" provided in Section 3.1.5 of [RFC4689].

Discussion:
A flow can be a set of packets having the same source address, destination address, source port, and destination port, or any combination of these items.

Measurement Units:
N/A

2.1.2. Northbound Interface

Definition:
The definition of "northbound interface" is the same as the definition of "service interface" provided in [RFC7426].

Discussion:
The northbound interface allows SDN applications and orchestration systems to program and retrieve the network information through the SDN Controller.

Measurement Units:
N/A

2.1.3. Southbound Interface

Definition:
The southbound interface is the application programming interface provided by the SDN Controller to interact with the SDN nodes.

Discussion:
The southbound interface enables the controller to interact with the SDN nodes in the network for dynamically defining the traffic forwarding behavior.

Measurement Units:
N/A

2.1.4. Controller Forwarding Table

Definition:
A controller Forwarding Table contains flow entries learned in one of two ways: first, entries can be learned from traffic received through the data plane, or second, these entries can be statically provisioned on the controller and distributed to devices via the southbound interface.

Discussion:
The controller Forwarding Table has an aging mechanism that will be applied only for dynamically learned entries.

Measurement Units:
N/A

2.1.5. Proactive Flow Provisioning Mode

Definition:
Controller programming flows in Network Devices based on the flow entries provisioned through the controller's northbound interface.

Discussion:
Network orchestration systems and SDN applications can define the network forwarding behavior by programming the controller, using Proactive Flow Provisioning. The controller can then program the Network Devices with the pre-provisioned entries.

Measurement Units:
N/A

2.1.6. Reactive Flow Provisioning Mode

Definition:
Controller programming flows in Network Devices based on the traffic received from Network Devices through the controller's southbound interface.

Discussion:
The SDN Controller dynamically decides the forwarding behavior based on the incoming traffic from the Network Devices. The controller then programs the Network Devices, using Reactive Flow Provisioning.

Measurement Units:
N/A

2.1.7. Path

Definition:
Refer to Section 5 in [RFC2330].

Discussion:
None

Measurement Units:
N/A

2.1.8. Standalone Mode

Definition:
A single controller handles all control-plane functionalities without redundancy, and it is unable to provide high availability and/or automatic failover.

Discussion:
In standalone mode, one controller manages one or more network domains.

Measurement Units:
N/A

2.1.9. Cluster/Redundancy Mode

Definition:
In this mode, a group of two or more controllers handles all control-plane functionalities.

Discussion:
In cluster mode, multiple controllers are teamed together for the purpose of load sharing and/or high availability. The controllers in the group may operate in active/standby (master/slave) or active/active (equal) mode, depending on the intended purpose.

Measurement Units:
N/A

2.1.10. Asynchronous Message

Definition:
Any message from the Network Device that is generated for network events.

Discussion:
Control messages like flow setup request and response messages are classified as asynchronous messages. The controller has to return a response message. Note that the Network Device will not be in blocking mode and continues to send/receive other control messages.

Measurement Units:
N/A

2.1.11. Test Traffic Generator

Definition:
The test traffic generator is an entity that generates/receives network traffic.

Discussion:
The test traffic generator typically connects with Network Devices to send/receive real-time network traffic.

Measurement Units:
N/A

2.1.12. Leaf-Spine Topology

Definition:
"Leaf-Spine" is a two-layered network topology, where a series of leaf switches that form the access layer are fully meshed to a series of spine switches that form the backbone layer.

Discussion:
In the Leaf-Spine topology, every leaf switch is connected to each of the spine switches in the topology.

Measurement Units:
N/A

2.2. Test Configuration/Setup Terms

2.2.1. Number of Network Devices

Definition:
The number of Network Devices present in the defined test topology.

Discussion:
The Network Devices defined in the test topology can be deployed using real hardware or can be emulated in hardware platforms.

Measurement Units:
Number of Network Devices.

2.2.2. Trial Repetition

Definition:
The number of times the test needs to be repeated.

Discussion:
The test needs to be repeated for multiple iterations to obtain a reliable metric. It is recommended that this test SHOULD be performed for at least 10 iterations to increase confidence in the measured results.

Measurement Units:
Number of trials.

2.2.3. Trial Duration

Definition:
Defines the duration of test trials for each iteration.

Discussion:
The Trial Duration forms the basis for "stop" criteria for benchmarking tests. Trials not completed within this time interval are considered incomplete.

Measurement Units:
Seconds.

2.2.4. Number of Cluster Nodes

Definition:
Defines the number of controllers present in the controller cluster.

Discussion:
This parameter is relevant when testing the controller's performance in clustering/teaming mode. The number of nodes in the cluster MUST be greater than 1.

Measurement Units:
Number of controller nodes.

2.3. Benchmarking Terms

This section defines metrics for benchmarking the SDN Controller. The procedure for performing the defined metrics is defined in the companion methodology document [RFC8456].

2.3.1. Performance

2.3.1.1. Network Topology Discovery Time

Definition:
The time taken by the controller(s) to determine the complete network topology, defined as the interval starting with the first discovery message from the controller(s) at its southbound interface and ending with all features of the static topology determined.

Discussion:
Network topology discovery is key for the SDN Controller to provision and manage the network, so it is important to measure how quickly the controller discovers the topology to learn the

current network state. This benchmark is obtained by presenting a network topology (tree, mesh, or linear) with a specified number of nodes to the controller and waiting for the discovery process to complete. It is expected that the controller supports a network discovery mechanism and uses protocol messages for its discovery process.

Measurement Units:
Milliseconds.

2.3.1.2. Asynchronous Message Processing Time

Definition:
The time taken by the controller(s) to process an asynchronous message, defined as the interval starting with an asynchronous message from a Network Device after the discovery of all the devices by the controller(s) and ending with a response message from the controller(s) at its southbound interface.

Discussion:
For SDN to support dynamic network provisioning, it is important to measure how quickly the controller responds to an event triggered from the network. The event can be any notification messages generated by a Network Device upon arrival of a new flow, link down, etc. This benchmark is obtained by sending asynchronous messages from every connected Network Device one at a time for the defined Trial Duration. This test assumes that the controller will respond to the received asynchronous messages.

Measurement Units:
Milliseconds.

2.3.1.3. Asynchronous Message Processing Rate

Definition:
The number of responses to asynchronous messages per second (a new flow arrival notification message, link down, etc.) for which the controller(s) performed processing and replied with a valid and productive (non-trivial) response message.

Discussion:
As SDN assures a flexible network and agile provisioning, it is important to measure how many network events (a new flow arrival notification message, link down, etc.) the controller can handle at a time. This benchmark is measured by sending asynchronous messages from every connected Network Device at the rate that the controller processes (without dropping them). This test assumes

that the controller responds to all the received asynchronous messages (the messages can be designed to elicit individual responses).

When sending asynchronous messages to the controller(s) at high rates, some messages or responses may be discarded or corrupted and require retransmission to controller(s). Therefore, a useful qualification on the Asynchronous Message Processing Rate is whether the incoming message count equals the response count in each trial. This is called the Loss-Free Asynchronous Message Processing Rate.

Note that several of the early controller benchmarking tools did not consider lost messages and instead report the maximum response rate. This is called the Maximum Asynchronous Message Processing Rate.

To characterize both the Loss-Free Asynchronous Message Processing Rate and the Maximum Asynchronous Message Processing Rate, a test can begin the first trial by sending asynchronous messages to the controller(s) at the maximum possible rate and can then record the message reply rate and the message loss rate. The message-sending rate is then decreased by the STEP size. The message reply rate and the message loss rate are recorded. The test ends with a trial where the controller(s) processes all of the asynchronous messages sent without loss. This is the Loss-Free Asynchronous Message Processing Rate.

The trial where the controller(s) produced the maximum response rate is the Maximum Asynchronous Message Processing Rate. Of course, the first trial can begin at a low sending rate with zero lost responses and then increase the rate until the Loss-Free Asynchronous Message Processing Rate and the Maximum Asynchronous Message Processing Rate are discovered.

Measurement Units:
Messages processed per second.

2.3.1.4. Reactive Path Provisioning Time

Definition:
The time taken by the controller to set up a path reactively between source and destination nodes, defined as the interval starting with the first flow provisioning request message received by the controller(s) and ending with the last flow provisioning response message sent from the controller(s) at its southbound interface.

Discussion:
As SDN supports agile provisioning, it is important to measure how fast the controller provisions an end-to-end flow in the data plane. The benchmark is obtained by sending traffic from a source endpoint to the destination endpoint and finding the time difference between the first and last flow provisioning message exchanged between the controller and the Network Devices for the traffic path.

Measurement Units:
Milliseconds.

2.3.1.5. Proactive Path Provisioning Time

Definition:
The time taken by the controller to proactively set up a path between source and destination nodes, defined as the interval starting with the first proactive flow provisioned in the controller(s) at its northbound interface and ending with the last flow provisioning command message sent from the controller(s) at its southbound interface.

Discussion:
For SDN to support pre-provisioning of the traffic path from the application, it is important to measure how fast the controller provisions an end-to-end flow in the data plane. The benchmark is obtained by provisioning a flow on the controller's northbound interface for the traffic to reach from a source to a destination endpoint and finding the time difference between the first and last flow provisioning message exchanged between the controller and the Network Devices for the traffic path.

Measurement Units:
Milliseconds.

2.3.1.6. Reactive Path Provisioning Rate

Definition:
The maximum number of independent paths a controller can concurrently establish per second between source and destination nodes reactively, defined as the number of paths provisioned per second by the controller(s) at its southbound interface for the flow provisioning requests received for path provisioning at its southbound interface between the start of the trial and the expiry of the given Trial Duration.

Discussion:
For SDN to support agile traffic forwarding, it is important to measure how many end-to-end flows the controller can set up in the data plane. This benchmark is obtained by sending each traffic flow with unique source and destination pairs from the source Network Device and determining the number of frames received at the destination Network Device.

Measurement Units:
Paths provisioned per second.

2.3.1.7. Proactive Path Provisioning Rate

Definition:
The maximum number of independent paths a controller can concurrently establish per second between source and destination nodes proactively, defined as the number of paths provisioned per second by the controller(s) at its southbound interface for the paths provisioned in its northbound interface between the start of the trial and the expiry of the given Trial Duration.

Discussion:
For SDN to support pre-provisioning of the traffic path for a larger network from the application, it is important to measure how many end-to-end flows the controller can set up in the data plane. This benchmark is obtained by sending each traffic flow with unique source and destination pairs from the source Network Device. Program the flows on the controller's northbound interface for traffic to reach from each of the unique source and destination pairs, and determine the number of frames received at the destination Network Device.

Measurement Units:
Paths provisioned per second.

2.3.1.8. Network Topology Change Detection Time

Definition:
The amount of time taken by the controller to detect any changes in the network topology, defined as the interval starting with the notification message received by the controller(s) at its southbound interface and ending with the first topology rediscovery messages sent from the controller(s) at its southbound interface.

Discussion:
In order for the controller to support fast network failure recovery, it is critical to measure how fast the controller is able to detect any network-state change events. This benchmark is obtained by triggering a topology change event and measuring the time the controller takes to detect and initiate a topology rediscovery process.

Measurement Units:
Milliseconds.

2.3.2. Scalability

2.3.2.1. Control Sessions Capacity

Definition:
The maximum number of control sessions the controller can maintain, defined as the number of sessions that the controller can accept from Network Devices, starting with the first control session and ending with the last control session that the controller(s) accepts at its southbound interface.

Discussion:
Measuring the controller's Control Sessions Capacity is important for determining the controller's system and bandwidth resource requirements. This benchmark is obtained by establishing a control session with the controller from each of the Network Devices until the controller fails. The number of sessions that were successfully established will provide the Control Sessions Capacity.

Measurement Units:
Maximum number of control sessions.

2.3.2.2. Network Discovery Size

Definition:
The network size (number of nodes and links) that a controller can discover, defined as the size of a network that the controller(s) can discover, starting with a network topology provided by the user for discovery and ending with the number of nodes and links that the controller(s) can successfully discover.

Discussion:
Measuring the maximum network size that the controller can discover is key to optimal network planning. This benchmark is obtained by presenting an initial set of Network Devices for discovery to the controller. Based on the initial discovery, the

number of Network Devices is increased or decreased to determine the maximum number of nodes and links that the controller can discover.

Measurement Units:
Maximum number of network nodes and links.

2.3.2.3. Forwarding Table Capacity

Definition:
The maximum number of flow entries that a controller can manage in its Forwarding Table.

Discussion:
It is important to measure the capacity of a controller's Forwarding Table to determine the number of flows that the controller can forward without flooding or dropping any traffic. This benchmark is obtained by continuously presenting the controller with new flow entries through the Reactive Flow Provisioning mode or the Proactive Flow Provisioning mode until the Forwarding Table becomes full. The maximum number of nodes that the controller can hold in its Forwarding Table will provide the Forwarding Table Capacity.

Measurement Units:
Maximum number of flow entries managed.

2.3.3. Security

2.3.3.1. Exception Handling

Definition:
To determine the effect of handling error packets and notifications on performance tests.

Discussion:
This benchmark is to be performed after obtaining the baseline measurement results for the performance tests defined in Section 2.3.1. This benchmark determines the deviation from the baseline performance due to the handling of error or failure messages from the connected Network Devices.

Measurement Units:
Deviation from baseline metrics while handling Exceptions.

2.3.3.2. Handling Denial-of-Service Attacks

Definition:
To determine the effect of handling denial-of-service (DoS) attacks on performance and scalability tests.

Discussion:
This benchmark is to be performed after obtaining the baseline measurement results for the performance and scalability tests defined in Sections 2.3.1 and 2.3.2. This benchmark determines the deviation from the baseline performance due to the handling of DoS attacks on the controller.

Measurement Units:
Deviation from baseline metrics while handling DoS attacks.

2.3.4. Reliability

2.3.4.1. Controller Failover Time

Definition:
The time taken to switch from an active controller to the backup controller when the controllers operate in redundancy mode and the active controller fails, defined as the interval starting when the active controller is brought down and ending with the first rediscovery message received from the new controller at its southbound interface.

Discussion:
This benchmark determines the impact of provisioning new flows when controllers are teamed together and the active controller fails.

Measurement Units:
Milliseconds.

2.3.4.2. Network Re-provisioning Time

Definition:
The time taken by the controller to reroute traffic when there is a failure in existing traffic paths, defined as the interval starting with the first failure notification message received by the controller and ending with the last flow re-provisioning message sent by the controller at its southbound interface.

Discussion:
This benchmark determines the controller's re-provisioning ability upon network failures and makes the following assumptions:

1. The network topology supports a redundant path between the source and destination endpoints.

2. The controller does not pre-provision the redundant path.

Measurement Units:
Milliseconds.

3. Test Setup

This section provides common reference topologies that are referred to in individual tests defined in the companion methodology document [RFC8456].

3.1. Test Setup - Controller Operating in Standalone Mode

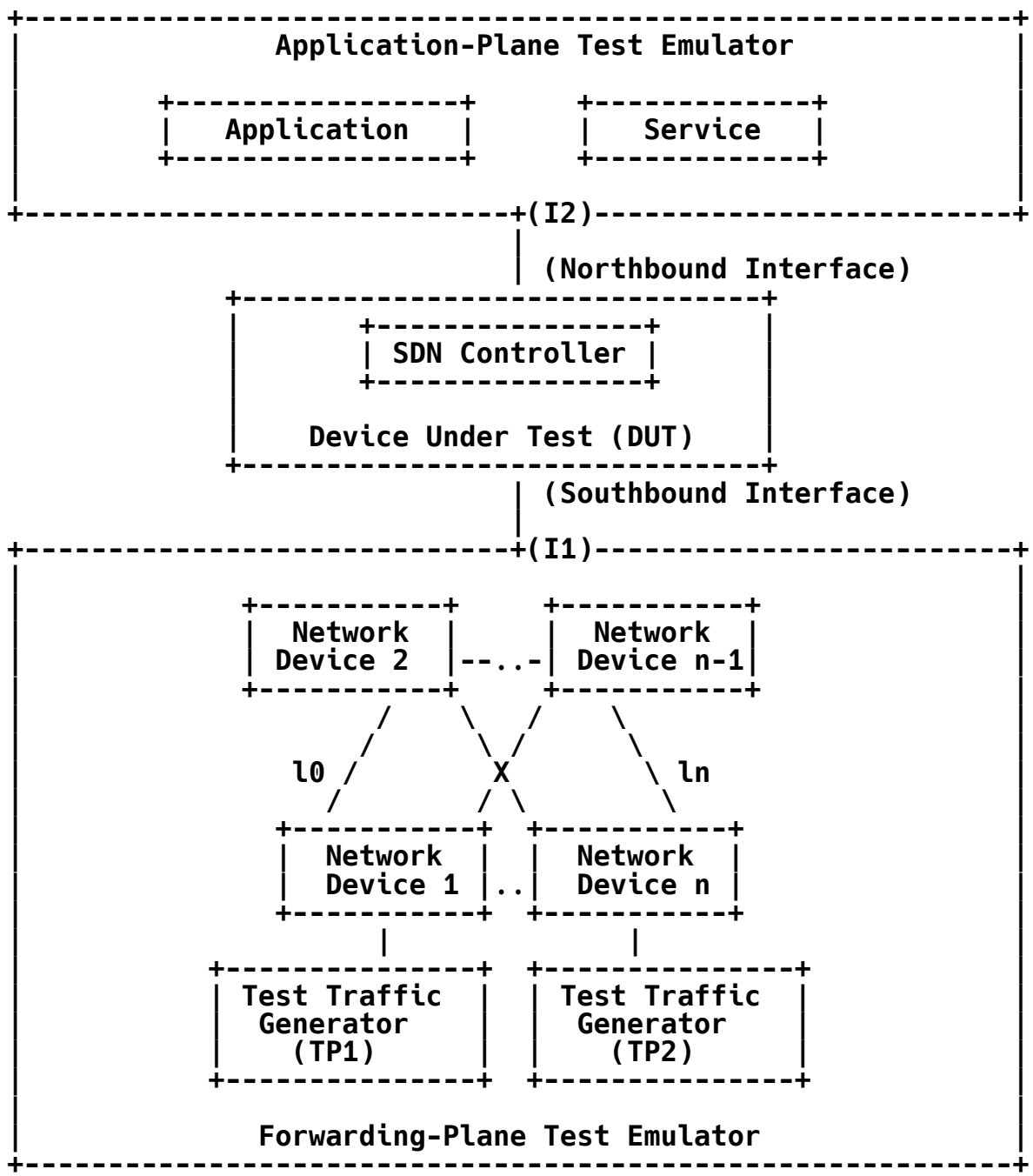

Figure 1

3.2. Test Setup - Controller Operating in Cluster Mode

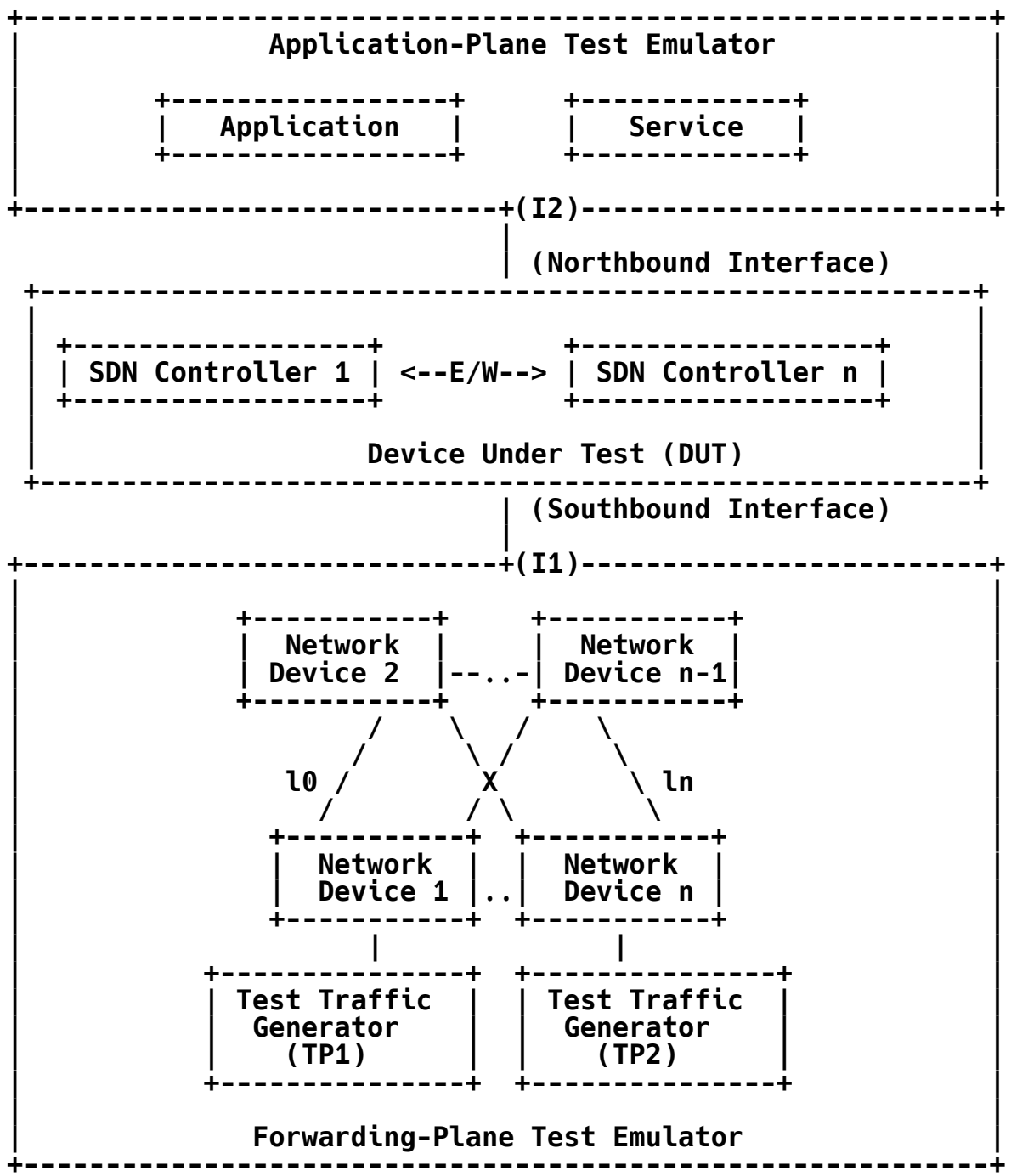

Figure 2

4. Test Coverage

Lifecycle	Speed	Scalability	Reliability
Setup	1. Network Topology Discovery Time 2. Reactive Path Provisioning Time 3. Proactive Path Provisioning Time 4. Reactive Path Provisioning Rate 5. Proactive Path Provisioning Rate	1. Network Discovery Size	
Operational	1. Maximum Asynchronous Message Processing Rate 2. Loss-Free Asynchronous Message Processing Rate 3. Asynchronous Message Processing Time	1. Control Sessions Capacity 2. Forwarding Table Capacity	1. Network Topology Change Detection Time 2. Exception Handling 3. Handling Denial-of-Service Attacks 4. Network Re-provisioning Time
Teardown			1. Controller Failover Time

5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

The benchmarking tests described in this document are limited to the performance characterization of controllers in a lab environment with isolated networks.

The benchmarking network topology will be an independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network or misroute traffic to the test management network.

Further, benchmarking is performed on a "black-box" basis, relying solely on measurements observable external to the controller.

Special capabilities SHOULD NOT exist in the controller specifically for benchmarking purposes. Any implications for network security arising from the controller SHOULD be identical in the lab and in production networks.

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <https://www.rfc-editor.org/info/rfc2330>.

[RFC4689] Poretsky, S., Perser, J., Erramilli, S., and S. Khurana, "Terminology for Benchmarking Network-layer Traffic Control Mechanisms", RFC 4689, DOI 10.17487/RFC4689, October 2006, <https://www.rfc-editor.org/info/rfc4689>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8456] Bhuvaneswaran, V., Basil, A., Tassinari, M., Manral, V., and S. Banks, "Benchmarking Methodology for Software-Defined Networking (SDN) Controller Performance", RFC 8456, DOI 10.17487/RFC8456, October 2018, <https://www.rfc-editor.org/info/rfc8456>.

Acknowledgments

The authors would like to acknowledge Al Morton (AT&T) for his significant contributions to the earlier draft versions of this document. The authors would like to thank the following individuals for providing their valuable comments to the earlier draft versions of this document: Sandeep Gangadharan (HP), M. Georgescu (NAIST), Andrew McGregor (Google), Scott Bradner, Jay Karthik (Cisco), Ramki Krishnan (VMware), and Boris Khasanov (Huawei).

Authors' Addresses

Bhuvaneswaran Vengainathan
Veryx Technologies Inc.
1 International Plaza, Suite 550
Philadelphia, PA 19113
United States of America

Email: bhuvaneswaran.vengainathan@veryxtech.com

Anton Basil
Veryx Technologies Inc.
1 International Plaza, Suite 550
Philadelphia, PA 19113
United States of America

Email: anton.basil@veryxtech.com

Mark Tassinari
Hewlett Packard Enterprise
8000 Foothills Blvd.
Roseville, CA 95747
United States of America

Email: mark.tassinari@hpe.com

Vishwas Manral
NanoSec Co
3350 Thomas Rd.
Santa Clara, CA 95054
United States of America

Email: vishwas.manral@gmail.com

Sarah Banks
VSS Monitoring
930 De Guigne Drive
Sunnyvale, CA 94085
United States of America

Email: sbanks@encrypted.net