ETSI GR NFV-TST 004 V1.1.2 (2017-07)

Network Functions Virtualisation (NFV); Testing; Guidelines for Test Plan on Path Implementation through NFVI

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGR/NFV-TST004ed112

Keywords
NFV, NFVI, SDN, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights 4
Foreword 4
Modal verbs terminology 4
Introduction 4
1 Scope 5
2 References 5
2.1 Normative references 5
2.2 Informative references 5
3 Definitions and abbreviations 7
3.1 Definitions 7
3.2 Abbreviations 7
4 Test Plan and Approach 8
5 Taxonomy of Options for SUT 9
6 Metrics and Methods of Measurement 10
7 Test Procedures 13
7.1 Prerequisites 13
7.2 Virtual Machine and VNF Instantiation 13
7.3 Network Preparation and Address Assignment 13
7.4 Path Instantiation 13
7.5 Path Performance 14
8 Results Presentation 14
9 Follow-on Activities 14
Annex A: Example of Prerequisites for Path Testing 15
A.1 Description and Figures of the System Under Test (SUT) 15
Annex B: Examples of Measurements and Results for Path Testing 19
B.1 Instantiation Time Measurements 19
B.2 Latency Measurements 20
B.2.1 Introduction 20
B.2.2 First Packet Latency 21
B.2.3 Subsequent Packet Latency 21
Annex C: Authors & contributors 24
History 25

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

There are many technological options available to implement paths through the Network Function Virtualisation Infrastructure (NFVI) to realize Virtual Network Forwarding Graphs (VNFFG) or Service Function Chains (SFC). In the present document, paths can be composed of physical and virtual links (including wide-area network links connecting locations and their NFVI), physical and virtual switches and routers, and other virtual network functions (VNF). VNFs are composed of one or more VNF Components (VNFC). VNFC are synonymous with Virtual Machines (VM) or OS containers (OSC) as in ETSI GS NFV 003 [i.21]. A VM or OSC is referred to with the general term virtualization container in the present document.

The present document is motivated by the design needs of many NFV actors. Service Providers and NFVI Operators need to select the best alternatives in order to implement cost-effective services. NFVI Providers and VNF Providers need to understand the preferred alternatives so they can support them efficiently. What configurations work well in combination, and possibly enhance performance? How can the actors above begin to objectively evaluate the various alternatives? These questions are to be evaluated before NFV deployments, and re-evaluated as new technology alternatives emerge.

The present document recognizes the need to evaluate the various path-implementation alternatives operating together, and begins by providing a high level test plan. Ultimately, the results from tests comparing the alternatives may influence the architectural choices when implementing the NFV framework.

1 Scope

The present document provides guidelines for test plans that assess different approaches to defining SDN Applications, different ways of arranging and federating SDN Controllers, and arrangements of network switching/forwarding functions (both physical and virtual) to create the various path-implementations between and among NS Endpoints and VNFs. These guidelines support development of detailed test plans, and ultimately influence the NFV framework (when testers share their results from testing arrangements encouraged by these guidelines). The test plan guidelines should be sufficiently abstract to include all envisioned possibilities, and will also pursue the details of technologies of interest. Although the primary emphasis of testing is the performance and benchmarking of systems composed of the components above, the attempts to combine different protocols and functions will undoubtedly uncover combinations which are non-interoperable, and these should be noted.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV-EVE 005 (V1.1.1) (2015-12): "Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework".

[i.2] IETF RFC 6241(June 2011): "Network Configuration Protocol (NETCONF)".

[i.3] ONOS™.

NOTE: Available at http://onosproject.org/.

[i.4] OpenDaylight™ .

NOTE: Available at http://www.opendaylight.org/.

[i.5] OpenContrail™.

NOTE: Available at http://www.opencontrail.org/.

[i.6] Floodlight™.

NOTE: Available at http://www.projectfloodlight.org/floodlight/.

[i.7] OpenStack™ SM .

NOTE: Available at https://www.openstack.org/.

[i.8] IETF RFC 4271 (January 2006): "A Border Gateway Protocol 4 (BGP-4)".

[i.9] IETF RFC 5440 (March 2009): "Path Computation Element (PCE) Communication Protocol (PCEP)".

[i.10] OpenFlow℠.

NOTE: Available at https://www.opennetworking.org/sdn-resources/openflow.

[i.11] P4™ language for programming the network dataplane.

NOTE: Available at http://p4.org/.

[i.12] ETSI GS NFV-INF 003 (V1.1.1) (2014-12): "Network Functions Virtualisation (NFV); Infrastructure; Compute Domain".

[i.13] VLOOP-VNF.

NOTE: Available at https://lists.linuxfoundation.org/pipermail/opnfv-tech-discuss/2015-May/002601.html.

[i.14] IETF RFC 7348 (August 2014): "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks".

[i.15] IETF RFC 7432 (February 2015): "BGP MPLS-Based Ethernet VPN".

[i.16] IETF RFC 1701 (October 1994): "Generic Routing Encapsulation (GRE)".

[i.17] Internet Draft (Work in Progress): "Geneve: Generic Network Virtualization Encapsulation, draft-ietf-nvo3-geneve-04".

[i.18] Internet Draft (Work in Progress): "Network Service Header, draft-ietf-sfc-nsh-11".

[i.19] Internet Draft (Work in Progress): "Benchmarking Methodology for SDN Controller Performance, draft-ietf-bmwg-sdn-controller-benchmark-meth-03".

[i.20] ETSI GS NFV-INF 010 (V1.1.1) (2014-12): "Network Functions Virtualisation (NFV); Service Quality Metrics".

[i.21] ETSI GS NFV 003 (V1.2.1) (2014-12): "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.22] ETSI GS NFV-TST 001 (V1.1.1) (2016-04): "Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services".

[i.23] IETF RFC 2544 (March 1999): "Benchmarking Methodology for Network Interconnect Devices".

[i.24] IETF RFC 2889 (August 2000): "Benchmarking Methodology for LAN Switching Devices".

[i.25] ETSI GS NFV-IFA 003 (V2.1.1) (2016-04): "Network Functions Virtualisation (NFV); Acceleration Technologies; vSwitch Benchmarking and Acceleration Specification".

[i.26] Internet Draft (Work in Progress): "Benchmarking Virtual Switches in OPNFV, draft-ietf-bmwg-vswitch-opnfv-01".

[i.27] Open Platform for NFV VSPERF Project.

NOTE: Available at https://wiki.opnfv.org/display/vsperf.

[i.28] IETF Benchmarking Methodology Working Group (BMWG).

NOTE: Available at https://datatracker.ietf.org/wg/bmwg/documents/.

[i.29] ETSI GS NFV-PER 001 (V1.1.2) (2014-12): "Network Functions Virtualisation (NFV); NFV Performance & Portability Best Practises".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

path: data communications feature of the system describing the sequence and identity of system components visited by packets, where the components of the path may be either logical or physical

NOTE: Examples of physical components include a physical switch or a network interface of a host, and an example of a logical component is a virtual network switch. Paths may be unidirectional or bi-directional. Paths may be further characterized as data plane or control plane when serving these classes of traffic, and as packet payload-agnostic or payload processing (as in the case of transcoding, compression, or encryption).

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACL	Access Control List
API	Application Programming Interface
BGP	Broder Gateway Protocol
BMWG	Benchmarking Methodology Working Group
CLI	Command Line Interface
EVPN	Ethernet Virtual Private Network
FUT	Function Under Test
GENEVE	Generic Network Virtualization Encapsulation
GR	Group Report
GRE	Generic Routing Encapsulation
GS	Group specification
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISG	Industry Specification Group
KVM	Kernel-based Virtual Machine
LTS	Long-Term Stability
MAC	Media Access Control
MB	Mega Bytes
NB	North Bound
NETCONF	Network Configuration Protocol

NOTE: See IETF RFC 6241 [i.2].

NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NIC	Network Interface Circuit
NS	Network Service
NSD	Network Service Description
NSH	Network Service Header
PCE	Path Computation Element
PCEP	PCE Communication Protocol

NOTE: See IETF RFC 5440 [i.9].

PNF	Physical Network Function
ODL	OpenDayLight
ONOS	Open Network Operating System
OSC	Operating System Container
OSS/BSS	Operation Support System/Business Support System
OSX	Apple Operating System for Mac
OVS	Open vSwitch

RPC	Remote Procedure Call
RTT	Round-Trip Time
SB	South Bound
SDN	Software Defined Network
SFC	Service Function Chains
SUT	System Under Test
VIM	Virtual Infrastructure Manager
VLOOP-VNF	Loopback Virtual Network Function
VM	Virtual Machine
VNF	Virtual Network Function
VNFC	VNF Component
VNFFG	VNF Forwarding Graph
VNFM	VNF Manager
VSPERF	OPNFV vSwitch Performance project
VXLAN	Virtual eXtensible Local Area Network

NOTE: See IETF RFC 7348 [i.14].

4 Test Plan and Approach

This clause outlines the first steps toward conducting a test of path instantiation and path performance.

The plan assumes that many selections of fundamental infrastructure have been made, such as the hardware platforms for compute, memory and storage, and hardware networking aspects such as switches, link technology and speed, and physical Network Interface Circuit (NIC) on each host. The configuration of these devices is a critical factor in their performance, and their parameters should be documented along with the tested technology-specific parameters (item 3 below). In principle, this allows evaluation of hardware alternatives, but this aspect is not emphasized beyond this point (consistent with the scope).

The plan also assumes that testing or benchmarking of individual components will be accessed or conducted in advance, as an aid to the selection of alternatives. After conducting tests according to this plan, it may be useful to re-examine component-level testing with the same or similar stimuli introduced in the path testing if the results are surprising or inconsistent, especially when the current instantiation differs from the benchmarked instantiation in an earlier test or different platform.

In the context of path testing, the System Under Test (SUT) consists of one or more Functions Under Test (FUT) and the network connecting the various FUT to establish the path itself.

The organization wishing to compare various path-implementation alternatives (candidates) and employing the guidance of the present document would build test cases as follows:

1) Determine the set of Functions Under Test (FUT) and the network connectivity that constitutes the path, including the physical arrangement of switches and hosts in each NFVI (and when there are more than one, the links between NFVI), the selected virtualization-layer, the availability of virtual functions and virtual switches, and the arrangement and configuration of SDN controller(s) along with their application-level and resource-level interfaces. The System Under Test (SUT) comprises all these components.

2) Determine the list of candidate data-plane and control-plane protocols and design of overlay networks (such as those given in clause 5).

3) Determine the complete set of configuration parameters required for repeatable results, and the range of configuration settings which the test runs will use to evaluate and compare the candidates.

4) Determine the test device configurations (test stimuli), as well as the metrics and benchmarks the test devices will collect (including intermediate metrics of the path instantiation process and segments of the end-to-end path), in addition to resource utilization reading/logging from the functions under test where appropriate.

5) Arrange to instantiate each combination of parameters, variables, protocols, function arrangements, and verify their operation.

6) Execute the resulting test cases and measure the selected metrics, and record the results.

7) Prepare a clear report of the results, sufficiently detailed to allow repeating the tests at a future date. This will usually include scripts prepared to automate the configuration, instantiation, and testing of the SUT.

When preferred combinations and implementations emerge in the analysis, the testing organization should ensure that the needed management capabilities are consistent with the NFV framework, or suggest how the framework might be modified to accommodate such implementations. The steps to evaluate management capabilities fall under the scope of interoperability testing and are beyond the present document's scope.

5 Taxonomy of Options for SUT

This clause organizes the options for the SUT in several categories, and provides examples of each category for clarity. The categories include Application-Control Interfaces and Protocols, SDN Controller type, Controller Arrangement with controller-controller protocols where necessary, Orchestration interfaces and protocols (direct to the SDN Controller), Resource Control interface(s) and protocols.

Figure 5 of ETSI GS NFV-EVE 005 [i.1] provides an illustration of the SDN controller interfaces, and provides terminology and organization to discuss the many options possible in the SUT.

Function Placement: Each of the functions described in figure 5 of ETSI GS NFV-EVE 005 [i.1] has Placement options in two categories - within the abstract NFV Reference Framework and within the Physical (and logical) resources of the SUT. Both the Framework and Physical placement will have performance implications in one or more of the metrics measured when testing path instantiation and path performance.

SDN Application Type: A control application could take several forms. One is Intent-based networking, where the desired network and VNF connectivity are expressed in a prescriptive manner (and many details are communicated in an abstract way). Another uses an exact description of packet-forwarding path outcome. The controller(s) may apply policies and acquired knowledge of network resources to fulfil the prescribed intent or exact description.

Application-Control interfaces: Sometimes called the "north-bound" (NB) interface of an SDN controller, this interface provides for communication between an SDN Application and the controller(s). Examples of this interface include RESTful APIs, Remote Procedure Call (RPC) interfaces, protocols such as NETCONF [i.2], inter-process communication, and others. The use of clear or secure protocols represents an additional option on this interface. The use of clear or secure protocols represents an additional option on any interface.

SDN Controller type: There are many different types of SDN Controllers available today, each with its own design strengths and features. ONOS [i.3], OpenDayLight® [i.4], OpenContrail [i.5], and FloodLight® [i.6] are a few of the active open-source controller projects. The SDN Controller and the VIM (such as OpenStack [i.7] Nova and Neutron) may both have a role in path setup.

Controller Arrangement and protocols: In some SUT designs, multiple controller entities may share the role of the SDN controller function, as indicated in figure 5 of ETSI GS NFV-EVE 005 [i.1]. The controllers are sometimes described as acting in a cluster, where one controller is the leader and others are followers, each having a partial view of the network resources and the paths under control. The protocols used between controllers (sometimes referred to as the east-west interface) may be provided by BGP IETF RFC 4271 [i.8] or other gateway protocols, plus alternatives such as the Path Computation Element (PCE) Communication Protocol (PCEP) [i.9] or OpenFlow [i.10].

Orchestration interfaces and protocols: This interface may be indirect or direct to the SDN Controller, and may re-use some of the protocols already listed, or other protocols and options yet to emerge.

Resource Control interface(s) and protocols: Sometimes called the SDN Controller Southbound (SB) interface, it provides communication between the controller(s) and the network functions under control, such as hardware switches and virtual switches. A commonly used protocol on this interface is Openflow [i.10], but others approaches are in development, such as the P4 language for programming the network dataplane [i.11]. The use of clear or secure protocols represents an additional set of options on this interface, when combined with the different choices of cypher-suites and acceleration technologies ETSI GS NFV-INF 003 [i.12].

Path Provisioning Models: The Resource Control Protocol may use a proactive or reactive provisioning model (or a combination of both). In the reactive model, flows are created asynchronously when packets arrive at the SDN Resource (switch) and the (first packet in the) flow's disposition is determined through a protocol exchange with the SDN Controller. A proactive model installs the necessary flows in tables at the SDN Resource (switch) before the flows arrive, a process which is usually conducted by the Controller under direction of an SDN Application over the Application-Control Interface.

SDN Resources: Although this plan assumes that many selections of fundamental infrastructure have been made, such as which switches will be implemented in hardware and which as virtual functions, the properties of these functions represent options which should be examined. Hardware networking options such as link technology and speed and physical Network Interface Circuit (NIC) on each host may be variable, along with the range of processing options presented by the resources/switches.

Virtual Network Functions: The path will include VNFs, and the type of VNF employed will determine the type of path testing possible and the applicability of the results. For example, a test VNF that simply returns traffic to the next link of the path/service chain/forwarding graph is one possibility and an open-source VLOOP-VNF is available to simplify performance testing [i.13]. Testing strategies may prefer to employ actual VNFs from the catalogue available to the testing organization, in which case the performance testing should measure the specifics of the network service composed by the path. VNFs implementing an Access Control List (ACL) are an option (see Annex A), although most forms of security-related testing are relegated to follow-on work.

Overlay Networking Technology: There are many options to conceal the end-to-end network addresses of packets and provide local direction and control by encapsulating the packets with new headers. Examples are VXLAN IETF RFC 7348 [i.14], EVPN IETF RFC 7432 [i.15], GRE IETF RFC 1701 [i.16], and GENEVE [i.17]. The use of clear or secure encapsulations represents an additional option for overlay networks. Network Service Header (NSH) [i.18] can be combined with overlay networks, and carries meta-data that can trigger actions that have an effect on performance, so this is another option.

6 Metrics and Methods of Measurement

This clause identifies and describes the key metrics for NFVI path performance, and describes the methods for measuring these metrics based on externally observable events.

There exist methods to characterize individual components of the path implementation architecture, such as SDN Controller benchmarking and virtual switch benchmarking ([i.19] and [i.26]). This clause assumes that individual components and resources in the SUT have been characterized to the extent desired, and that the traffic volumes generated in the tests described here will take the empirical limits discovered in such testing as inputs. For example, Network Resource Discovery (the map of controlled resources and their connectivity is an SDN Controller Benchmark [i.19].

ETSI GS NFV-INF 010 [i.20] provides performance metrics for instantiation of some components of the SUT, such as Virtual Machines and Virtual Network connectivity, as well as the performance of these components. Because the length of VNF instantiation may dominate the other time intervals defined and measured below, this key metric is defined and measured as a precursor to network-related metrics and measurements. The possibility exists to re-use the VNFC s and VNFs when testing different combinations of network technologies and techniques, and this would save considerable time between measurements. Therefore, **VM Provisioning Latency** defined in ETSI GS NFV-INF 010 [i.20] is a separate metric for the SUT, and was intended to be applicable to both VM and OSC instantiation where applicable. However, this metric is re-named **VNFC Instantiation Time** in the present document, to maintain independence from virtualization technology. See the definition of VNF Instance in ETSI GS NFV 003 [i.21], which includes many aspects that could be described as "provisioning".

Among this metric's key input parameters is the size of the image which is to be instantiated as a VNFC. The image size may be directly proportional to instantiation time in some systems. Other factors include the location of the requested image w.r.t. the VNFC instance, network load and host load.

VNFC Instantiation Time Definition: The time interval from the transmission of the request to instantiate the VNFC (to the VIM), to the time that the (remote) communication with the VNFC can be established (and full normal operation can be subsequently confirmed). The example method for measuring this metric is tabulated below.

Table 6.1: VNFC Instantiation Time Illustration and Example Operations

Operations	Description	OpenStack Nova Compute API or CLI example	Event Time Identification	Reference
1) Request (from VIM user or Orchestration)		Create Server	T1	http://developer.openstack.org/api-ref/compute/?expanded=create-server-detail
2) Monitor status of Request	Command to poll status of instances	nova list or GET /servers/{id}		http://docs.openstack.org/cli-reference/nova.html
3) Continue with Server Configuration	Commands to configure network, keys, allow remote access	nova get-vnc-console nova secgroup-add-rule		(both above)
4) Establish Communication	Command to setup secure communication	ssh user@serverIP	T2	http://docs.openstack.org/cli-reference/nova.html
5) Measurement	Calculate VNFC Instantiation Time		T2 - T1	ETSI GS NFV-INF 010 [i.20]
	Here, additional VNFCs are instantiated if needed, likely in parallel, along with intra-VNF communications			
6) Begin communication attempts to confirm VNF-specific operations	Verify application-specific functions which vary with the role of the VNF			
7) Successful VNF-specific communications confirmed	Note that VNF Indicators, as described in clause 7.11 of ETSI GS NFV-INF 010 [i.20], are values generated by the VNF itself, and therefore need external confirmation as shown here		T3	
8) Measurement	Calculate VNF Instantiation Time		T3 - T1	

Each VNF is composed of one or more VMs that host VNFCs. The **VNF Instantiation Time** includes the time to instantiate all component VNFCs and the connectivity between VNFCs. The method to identify and actuate connectivity between VNFCs is application-specific.

VNF Instantiation Time Definition: The time interval from the transmission of the request (to the VIM) to instantiate one or more VNFCs that comprise the VNF, to the time that VNF-specific communications can be conducted successfully and full normal operation can be subsequently confirmed. Table 6.1 includes these additional steps as part of the example.

The measurements are repeated for each VNF, measuring the overall time for VNFs composed of multiple VNFCs.

A key metric for the SUT is the Path Instantiation Time.

Path Instantiation Time Definition: The interval from the beginning of instantiation of the first of the following applicable path components and features:

- Virtual switches

- Network interfaces (physical and virtual)
- Network Links (physical and virtual)
- Establish SDN controller connectivity (NB and SB)
- Establish Layer 2 connectivity for all path components
- Establish higher-layer and overlay network connectivity for all relevant path components
- Completion of needed SDN controller communications (with application, switches, other controllers), including time to interpret the abstract SDN application prescription as specific commands for the available network resources and the provisioning of the flows across all switches in the particular path where applicable
- (plus the First-Packet Latency for both proactive and reactive provisioning models)

to the time that the complete path (dataplane) has been instantiated. This time may begin before all VNFs are instantiated, and path component instantiation and configuration processes may proceed in parallel where possible.

The First-Packet Latency contributes to Path Instantiation Time because it is a required component of the reactive provisioning model, and the complete path components may use both proactive and reactive provisioning (in the case that the complete path is provisioned proactively, the First-Packet Latency is unlikely to be a significant factor in the results).

It is possible to observe both the control-plane and data-plane components of Path Instantiation Time, where the control-plane may be verified by executing management commands, and the data-plane verified through connectivity testing. See Annex B for an example of Path Instantiation Time measurement.

NOTE: The length of VNF instantiation time alone may dominate the Path Instantiation Time on its own. Therefore, it may be necessary to pre-instantiate such VNFs, and explicitly exclude them from this metric, or to use test VNFs (VLOOP-VNF) whose instantiation time can be much less than a more complicated VNF.

One of the key metrics for paths through the infrastructure is the First-Packet Latency for new flows.

First-Packet Latency Definition: The round-trip latency of the first packet of a flow, measured between the end points of the path.

This is a bi-directional measurement because of independent provisioning of components on each direction of the path. In cases where the networking functions classify new packets and flows on entry and determine their next hop dynamically (as done in the reactive provisioning model), the processing time for the first packet of a flow is usually much greater than the subsequent packets in the same flow. The additional first-packet latency grows with each subsequent independent classification and forwarding decision. If the latency reaches the threshold where the end-user's application considers the first packet lost, the retransmission may be considered a new flow by the network classifiers, resulting in even more stress on the classification system, and unavailable communications to the user. See Annex B for an example of First Packet Latency measurement.

Once the path is established, then **Standard Packet Transfer Performance Metrics** (loss, delay, delay variation, reordering, duplication) can adequately describe the path performance for flows that are treated relatively transparently by the VNFs. For the case of Test VNFs such as VLOOP-VNF, these metrics apply to all packets. Since packet transfer performance may vary with time and be subject to the effects of resource utilization for other services and lifecycle-activities (such as neighbour VM instantiation or migration), longer duration tests are indicated for the best-performing combinations of SUT alternatives. These so-called soak-tests collect the same metrics as short duration tests, and may organize result collection so that time intervals with unusual performance (such as outlier delays or concentrated packet loss) can be isolated and examined in detail. See Annex B for an example of Packet Transfer Performance measurement.

A further consideration is that links and communications between entities on the path may be a first step in establishing connectivity for an overlay network that supports a complete Network Service or VNFFG, and it may be useful to have measurements on both the underlay network and overlay path instantiation, when used.

Repeated trials of any metrics should be summarized using the mean, variance (and other relevant statistics where desired).

7 Test Procedures

7.1 Prerequisites

The main prerequisite is to decide the goal(s) of the testing, in terms of the configurations whose performance will be measured and compared through the results generated here. These are steps 1 through 4 in clause 4 on Test Approach. Physical networking arrangements are prepared in this step.

The combination of VMs and network connectivity should be illustrated. In some cases, the SUT will form a Network Service (NS) ETSI GS NFV 003 [i.21]. Two categories of figures illustrating test setups should be included, one emphasizing the interconnection between functional components (in the abstract reference architecture), and another emphasizing physical placement in the SUT Infrastructure. These two categories can be combined if the resulting figure retains the needed clarity.

For the purposes of automating test setup and subsequent execution of measurements, the key aspects of the SUT may be viewed and specified as a NS, including the supporting test functions and their connectivity to/within the SUT, and following the full lifecycle of development, ingestion, instantiation and removal.

If a cloud platform or VIM and VNFM will perform some of the steps of this procedure, then another prerequisite is to install the VIM and verify its operation. If an SDN Controller is part of the VIM, then it is also instantiated at this step.

7.2 Virtual Machine and VNF Instantiation

The first step is to instantiate the needed VMs with the VNF image(s) required.

The Virtual Infrastructure Manager (VIM) is responsible for this operation. Therefore, the specific instructions for the VIM are needed to complete this operation.

If an SDN Controller is part of the SUT, then it is also instantiated at this step and the time to instantiate the controller should be measured according to the definition of VM provisioning (if applicable, controller placement may be in PNF, VNF, NFVI, VIM or OSS/BSS according to ETSI GS NFV-EVE 005 [i.1]).

Measurements: VNFC Instantiation Time, see ETSI GS NFV-INF 010 **[**i.20**]** and clause 6, and **VNF Instantiation Time**.

Verification: Clauses 6 and 7 of ETSI GS NFV-TST 001 [i.22] describe VNF instantiation verification tests.

7.3 Network Preparation and Address Assignment

Virtual networking arrangements are prepared in this step. This includes the establishment of Layer 2 broadcast domains, Layer 3 Subnets, establishment of Ports on the Subnets and attachment of VM interfaces to Ports (with MAC and IP address assignments). This step includes the configuration and/or creation of networking functions to support overlay networks and other special functions.

The VIM is capable of performing most aspects of this step, although additional configuration may be needed.

7.4 Path Instantiation

At this point, the necessary network capabilities and identities are instantiated and the procedure can begin to establish connectivity according to the figures/illustrations prepared to describe the SUT.

The Path Provisioning Model in use determines the next steps, when a single provisioning model is in-use (otherwise the combination of these steps is performed on each of the corresponding components of the path). Verification needs to be performed in all cases.

With a Proactive provisioning model, the set of flows required in the switches should be designed and formatted for the API of the SDN controller's application interface (North Bound, NB). The set of flows are then installed in the controller, and the controller's process to install the flows in the switches (South Bound, SB) commences and completes before verification below.

Measurement: (**Virtual) Proactive Network Provisioning Latency** ETSI GS NFV-INF 010 **[**i.20**]**, includes the time to program and install all flows, plus the First-Packet Latency.

With a Reactive provisioning model, the set of flows required in the switches need be established by sending traffic between functions on the path(s) of the NS. The switches typically inform the controller of the need to establish each new flow, and the controller responds to install the flows in the switches (South Bound, SB) and completes before verification below.

Measurement: (**Virtual) Reactive Network Provisioning Latency** ETSI GS NFV-INF 010 **[**i.20**]**, includes the time to identify and install all flows, plus the First-Packet Latency.

Note that First-Packet Latency measurements need not be conducted immediately following the instantiation of path components and activation of path features through configuration.

Verification: Verify that the necessary flows and connectivity between test functions and VNFs have been achieved, also that any unwanted connectivity is blocked by the network design.

7.5 Path Performance

Measurements of Path Performance can be categorized by the functions performed by the SUT on the traffic.

One category of SUT is the switching or routing category, where packets that pass through the SUT are unmodified in significant ways. The typical performance testing for this category is based on IETF RFC 2544 [i.23], IETF RFC 2889 [i.24] and the extensions described in ETSI GS NFV-TST 00 [i.22], ETSI GS NFV-IFA 003 [i.25], [i.26] and [i.27].

Another category is where the SUT performs some specific set of functions on the test traffic, such as when some processing is performed on the payload of test packets. The result of such processing may be a revised payload, or a corresponding response packet as in the case of control plane protocols. Performance testing for this category is based on the benchmarks described in [i.25], [i.26] and [i.27].

Additional tests may be conducted according to clause 7 of [i.22], and the specifications of the IETF Benchmarking Methodology Working Group (BMWG) [i.28].

8 Results Presentation

This clause points to examples for the presentation of results, both tabular and graphical.

The main goal of presenting the results is to simplify the comparison of alternatives tested, and illustrate the performance advantages or detractors of each alternative.

Annex B provides both tabular and graphical examples of results presentation for the experiment described there.

ETSI GS NFV-PER 001 [i.29], Annexes C and D also provide useful examples of result presentation and other aspects of test design.

9 Follow-on Activities

This clause provides a listing of the ideas for related work which were deemed beyond the scope of the present document.

Several areas of follow-on testing are anticipated, as described below:

- The Reliability aspects of each SUT should be evaluated for single function and single link failures, at least. Metrics will include recovery time to normal operation (for the complete SUT), recovery time to full capacity, and individual datapath outage times (which may be a range of times, with some paths restored before others).
- The Security aspects of each SUT should be evaluated for different attack vectors.

Annex A:
Example of Prerequisites for Path Testing

A.1 Description and Figures of the System Under Test (SUT)

As described in clause 7, several types of figures are required to clearly illustrate the path through the SUT. An example of each type of diagram is given below.

The goal of this set of tests is to evaluate ways to decompose the network functions typically found in physical network routers with dedicated hardware. The routing network function and the access control functions are separated and implemented independently. Network Service Description 1(NSD_1) in figure A1 illustrates a routing VNF and two Access Control List (ACL) VNFs, interconnected with the service endpoints and each other by virtual links.

The following SUT characteristics are fixed for this study:

- Virtualization Layer
- Virtual Router Forwarding Function
- Networking Overlay Protocols (none, L2 and L3 only)
- SDN Controller (when applicable)
- SDN Application (when applicable)

This study evaluates some alternatives for placement of the ACL functions as listed below:

1) ACL in VNF, vSwitch (Kernel and OVS) internal control for L2 switching, Shared host with router function.
2) ACL in VNF, vSwitch with independent SDN control , Shared host with router function.
3) ACL in vSwitch with independent SDN control and ACL SDN Application, Shared host with router function.
4) ACL in VNF, vSwitch with independent SDN control, ACL VNFCs in separate hosts from router function.
5) ACL in vSwitch with independent SDN control and ACL SDN Application, ACL in vSwitches in separate hosts from router function.

Each of these path implementation alternatives are illustrated in figures below, according to their list numbers above (however, only alternatives 1 through 3 will be examined here due to resource constraints).

The networking path components selected for testing the alternatives above are as follows:

- Independent SDN control: OpenDayLight® (ODL) Beryllium Release, SR4, karaf distribution with the following features loaded:
  - odl-restconf-all odl-mdsal-apidocs odl-dlux-all odl-l2switch-switch-ui;
  - OpenFlow 1.3 Network Resource control (south bound) interface.
- ACL SDN application: Network Intent Composition (NIC) with ODL Beryllium Release, SR4, embedded in controller or RESTconf application (north bound) interface.
- vSwitch: OVS version 2.0.2 and Kernel version (mininet 2.1.0+).
- VIM and virtualization layer: mininet 2.1.0+.

The following NFVI configuration parameters are fixed for this testing:

Table A.1: NFVI Configuration Parameters

Configuration Parameter	Value
Host Processor	2,5 GHz Intel Core i7
Dedicated Cores	2 (shared for all VM resources)
Memory	8 192 MB (1 600 MHz DDR3)
Acceleration	KVM Paravirtualization, VT-x, Nested Paging
OS	Ubuntu 14.04 LTS 64 bit (Guest in Virtual Box 5.x on Mac OSX 10.11.5)

The following metrics will be collected for each alternative test configuration:

- VNF Instantiation Time
- Path Instantiation Time
- First Packet Latency
- Standard Packet Transfer Performance Metrics (loss, delay, delay variation)

Provisioning Latency for the SDN Controller and VIM are desirable metrics to measure, where applicable.

It is also helpful to illustrate the packet headers at key points in the SUT, especially when overlay networking and encapsulation is used. In this example, all data path packets use IP and Ethernet headers, and so the figures are omitted for simplicity.

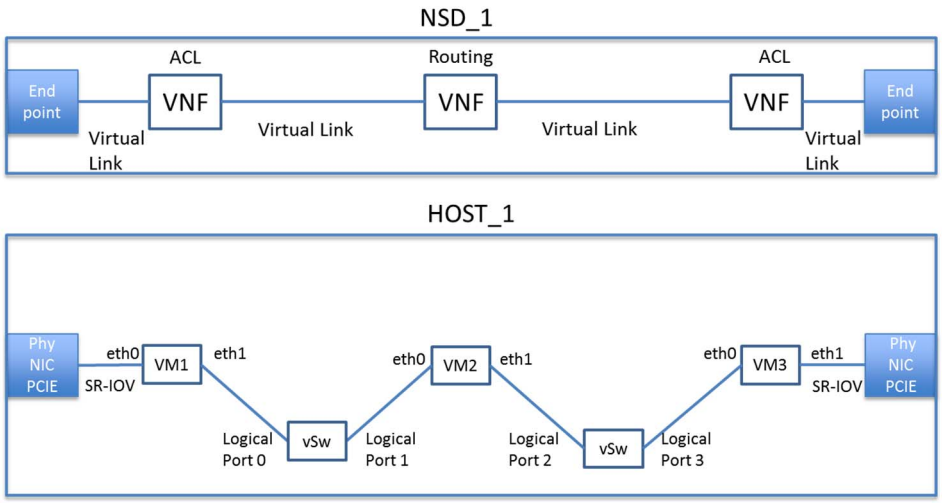


Figure A.1: ACL in VNF, vSwitch (Kernel and OVS) internal control for L2 switching, shared host with router function

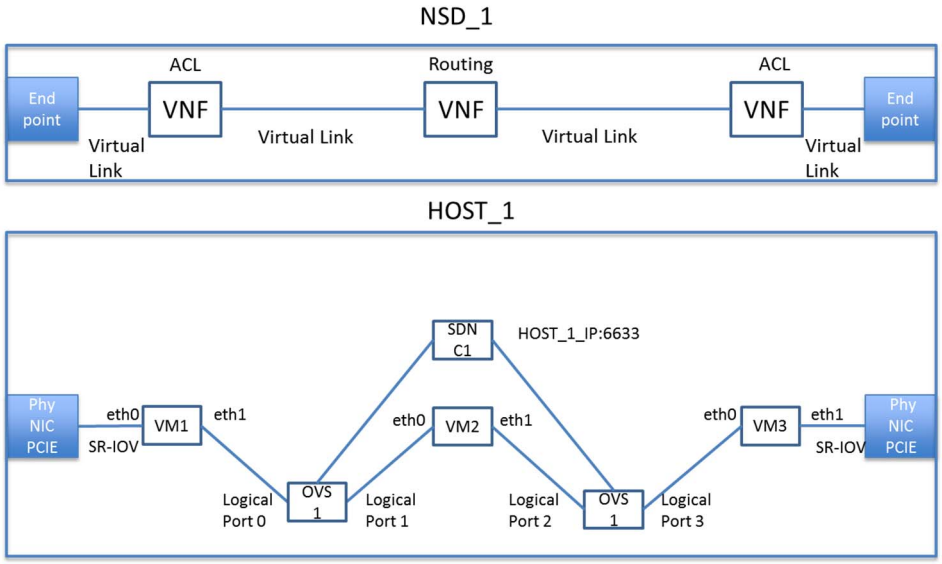


Figure A.2: ACL in VNF, vSwitch with independent SDN control, shared host with router function

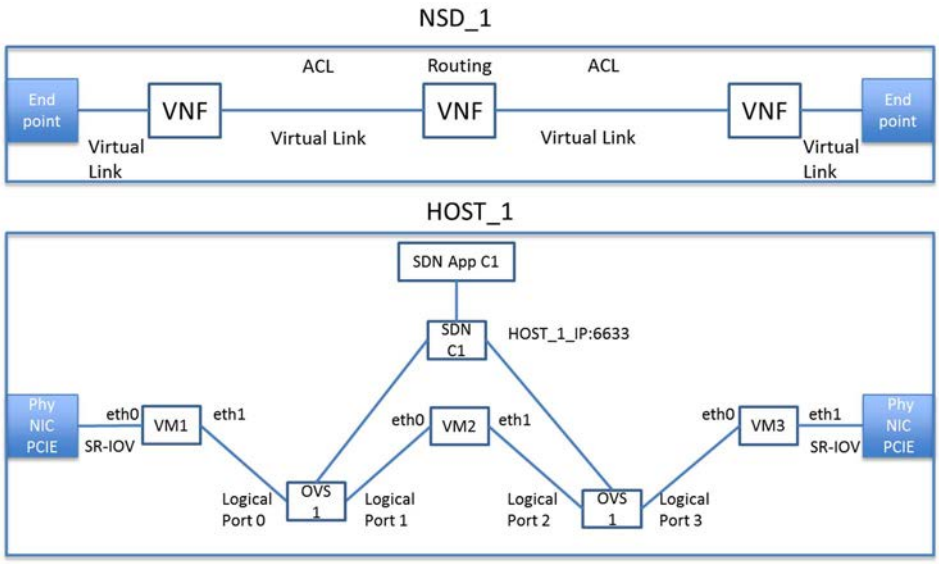
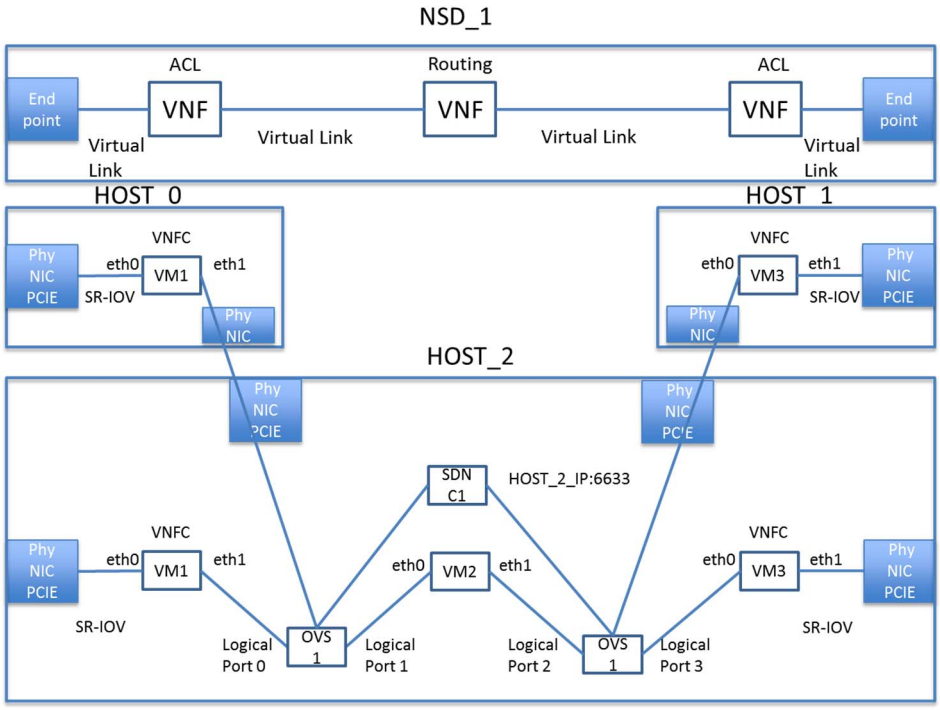
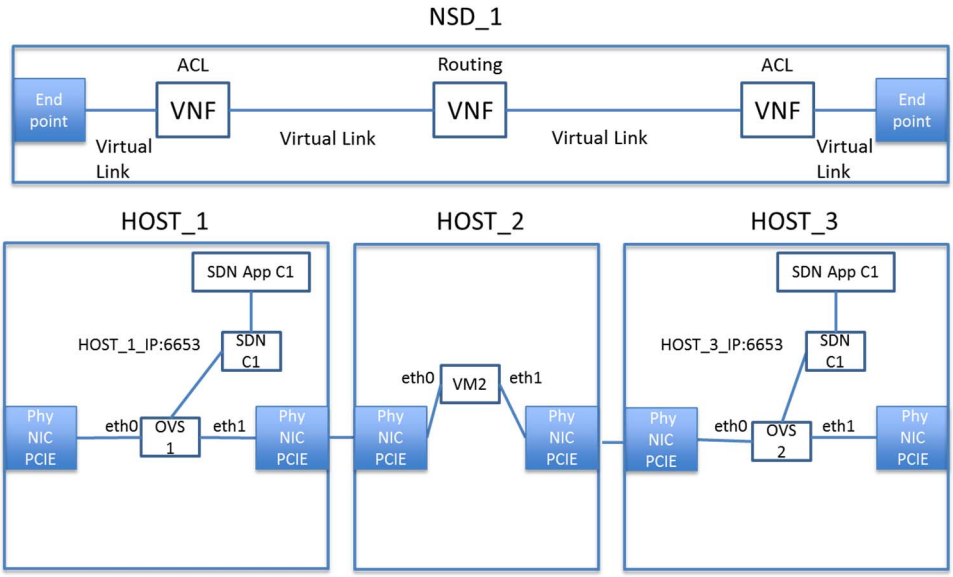


Figure A.3: ACL in vSwitch with independent SDN control and ACL SDN Application, shared host with router function



NOTE 1: ToR Switches provide connectivity between hosts.
NOTE 2: Either a cross-connect cable or physical switch provides connectivity between Hosts (not shown).

Figure A.4: ACL in VNF, vSwitch with independent SDN control, ACL VNFCs in separate hosts from router function



NOTE 1: ToR Switches provide connectivity between hosts.
NOTE 2: Either a cross-connect cable or physical switch provides connectivity between Hosts (not shown).

Figure A.5: ACL in vSwitch with independent SDN control and ACL SDN Application, ACL in vSwitches in separate hosts from router function

Annex B:
Examples of Measurements and Results for Path Testing

B.1 Instantiation Time Measurements

As described in clause 7, clear presentation of the results adds value to the study.

This experiment was conducted using mininet as the VIM. mininet uses process virtualization to create network resources (network namespaces) and hosts, which is similar to Linux containers but with fewer features. As a result VNF and Path Instantiation times are extremely fast with mininet, but all aspects of mininet share the compute environment with their host.

The steps to establish the figure A.2 path with OVS and ODL controller are shown below as an example of the procedure (which could be scripted for automation, as mentioned in clause 4, item 7):

The line below adds the features to the ODL controller, once started (with ./bin/karaf), and completion of this step comprises the SDN Controller Instantiation Time:

```
  opendaylight-user@root>feature:install odl-restconf-all odl-mdsal-apidocs
odl-dlux-all odl-l2switch-switch-ui
```

The line below adds the features to the ODL controller, once started (the completion of the host creation in this step constitutes the VNF Provisioning Latency):

```
  acm@al-ubuntu-VM:~$ sudo mn --mac --switch=ovsk,protocols=OpenFlow13 --
controller=remote --custom ~/mininet/custom/router.py --topo router
```

mininet starts, then configure IP addresses and hosts with source script (the completion of the networking portion of these two steps constitutes the Path Instantiation Time):

```
  mininet> source /home/acm/mininet/router_mn_scr_F1.txt
```

where the source script file above contains the following mininet commands:

```
  h1 ifconfig h1-eth0 192.168.12.1 netmask 255.255.255.0

  h2 ifconfig h2-eth0 192.168.12.2 netmask 255.255.255.0

  h2 ifconfig h2-eth1 192.168.23.2 netmask 255.255.255.0

  h3 ifconfig h3-eth0 192.168.23.3 netmask 255.255.255.0

  h1 route add default gw 192.168.12.2

  h3 route add default gw 192.168.23.2

  h2 sysctl net.ipv4.ip_forward=1

  h3 python -m SimpleHTTPServer 80 &
```

At this point, connectivity is verified and ICMP echo tests are run to measure latency with no ACL:

```
  mininet> h1 ping -c 10 h3
```

Then, the ACLs are added and ICMP echo tests are repeated (after verifying that port 80 is unreachable):

```
  mininet> h1 iptables -A OUTPUT -p tcp -d 192.168.23.3 --dport 80 -j DROP

  mininet> h3 iptables -A INPUT -p tcp -d 192.168.23.3 --dport 80 -j DROP

  Serving HTTP on 0.0.0.0 port 80 ...

  mininet> h1 ping -c 10 h3
```

Although most of the steps are similar for the figure A.3 Path, the network intent statement is shown below:

```
opendaylight-user@root>intent:add -f 00:00:00:00:00:01 -t 00:00:00:00:00:03
-a BLOCK

Intent created (id: 9f7b1a8b-f8c8-479e-91ce-5afa71287896)
```

Table B.1: Instantiation Time Measurements

Performance Metric	Measured Value
VNF Instantiation Time (3 VMs)	~100 msec
Component instantiation and configuration part of Path Instantiation Time (switches, links and host networking configurations)	~500 msec (add First-Packet Latency for total Path Instantiation Time, see table B.2)
ODL SDN Controller Instantiation Time (karaf distrib.)	~6 seconds

Verification of the path and its functions is a key aspect of the detailed procedures. Path connectivity was verified with and without ACLs installed during the procedure.

B.2 Latency Measurements

B.2.1 Introduction

As described in Annex A of the present document, various packet transfer Latency metrics are part of this study. The Latency measurements reported here are Round-Trip Time (RTT) measured using the simple ICMP Echo Request/Reply supported in ping tools. There were no packet losses observed at any time in the testing.

Figure B.1 illustrates the measurement path, and the locations of additional processes and path modifications needed to conduct the measurements, which also help to simplify the comparisons between alternatives.

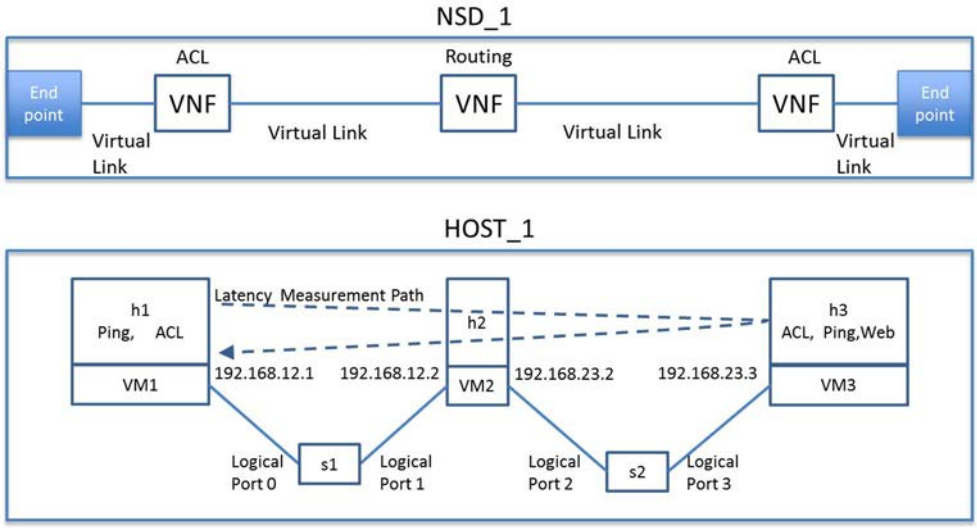


Figure B.1: NS and Measurement Path
(with modifications to simplify measurement and comparison)

Note that a simple web server is hosted in "h3". The ACLs applied block access to the webserver, but allow ICMP Echo packets to flow on the path (to allow the simple measurement of latency). In practice, an organization might choose to block the ICMP traffic instead. When h1 and h3 implement the ACL function, they each enforce one rule on their OUTPUT and INPUT chains, respectively.

B.2.2 First Packet Latency

For each of the tested path implementation alternatives, table B.2 summarizes the First Packet latency measurements.

Table B.2: First-Packet Latency Measurements

Path Alternative	A-1, Kernel Sw	A-1, OVS Switch	A-2, OVS Sw, ODL	A-3, OVS, ODL, ACLs in Switch
No ACL, RTT in msec	3,55	2,03	0,709	0,185
ACL Block Port 80, RTT in msec	1,33	1,54	0,328	
ACL Block MAC combo, RTT in msec				0,285

Note that it was not possible to implement the transport layer ACL in the L2 switches using Network Intent Composition, but a blocking rule was implemented on MAC address pairs.

B.2.3 Subsequent Packet Latency

For each of the tested path implementation alternatives, table B.3 summarizes the Packet latency measurements, excluding the first packet.

Table B.3: Subsequent Packet Latency Measurements

Path Alternative	A-1, Kernel Sw	A-1, OVS Switch	A-2, OVS Sw, ODL	A-3, OVS, ODL, ACLs in Switch
No ACL, RTT in msec Median (Std Dev)	0,075 (0,647)	0,064 (0,431)	0,055 (0,012)	0,051 (0,005)
ACL Block Port 80, RTT in msec, Median (Std Dev)	0,057 (0,633)	0,090 (0,364)	0,055 (0,004)	
ACL Block MAC combo, RTT in msec Median (Std Dev)				0,165 (0,113)

Figures B.2 through B.5 provide a time-series plot of subsequent packet round trip time (RTT) for each path alternative (note that the First Packet Latency is not plotted). In general, better and more consistent latency performance is seen when using the ODL SDN controller as opposed to the internal controller.

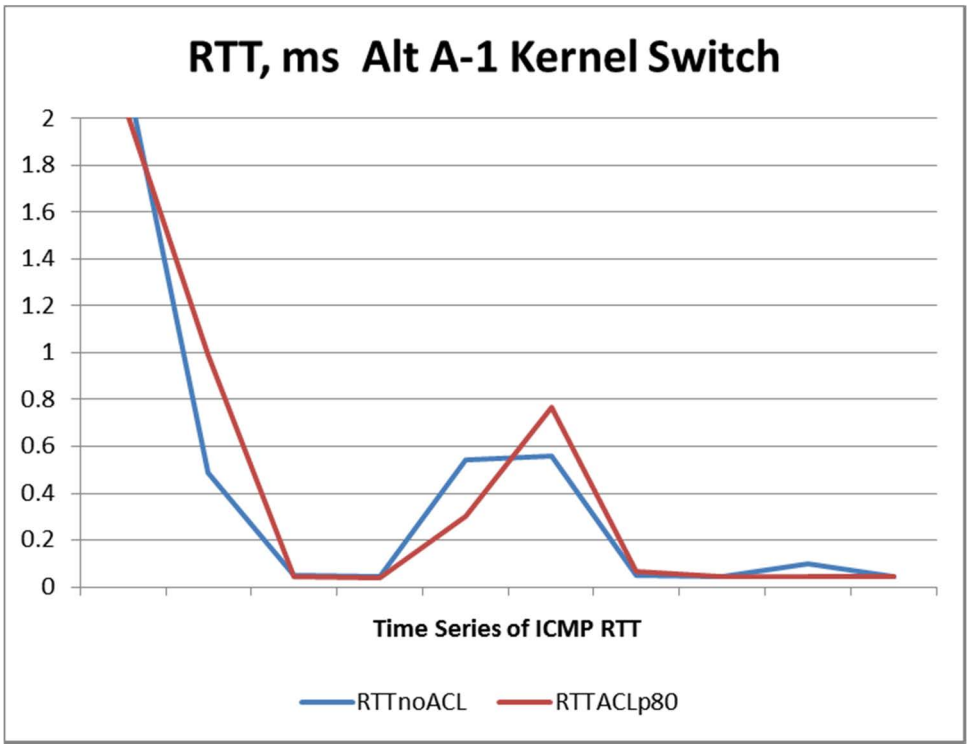


Figure B.2: Alternative A-1 Kernel Switch, Internal Controller, ACLs to restrict Port 80

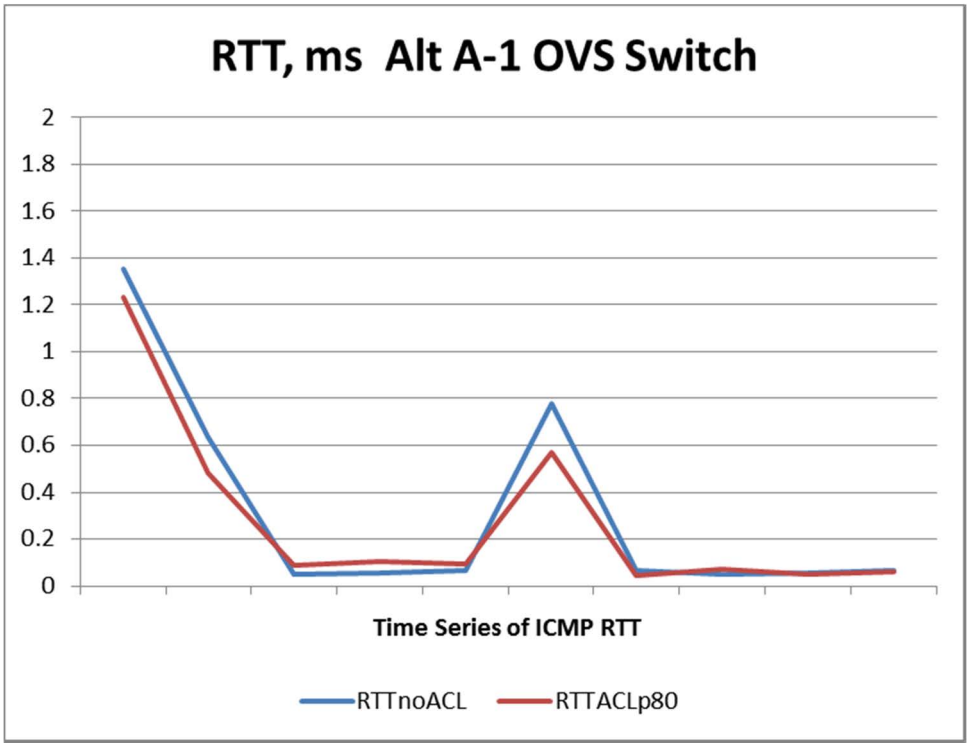


Figure B.3: Alternative A-1 OVS Switch, Internal Controller, ACLs to restrict Port 80

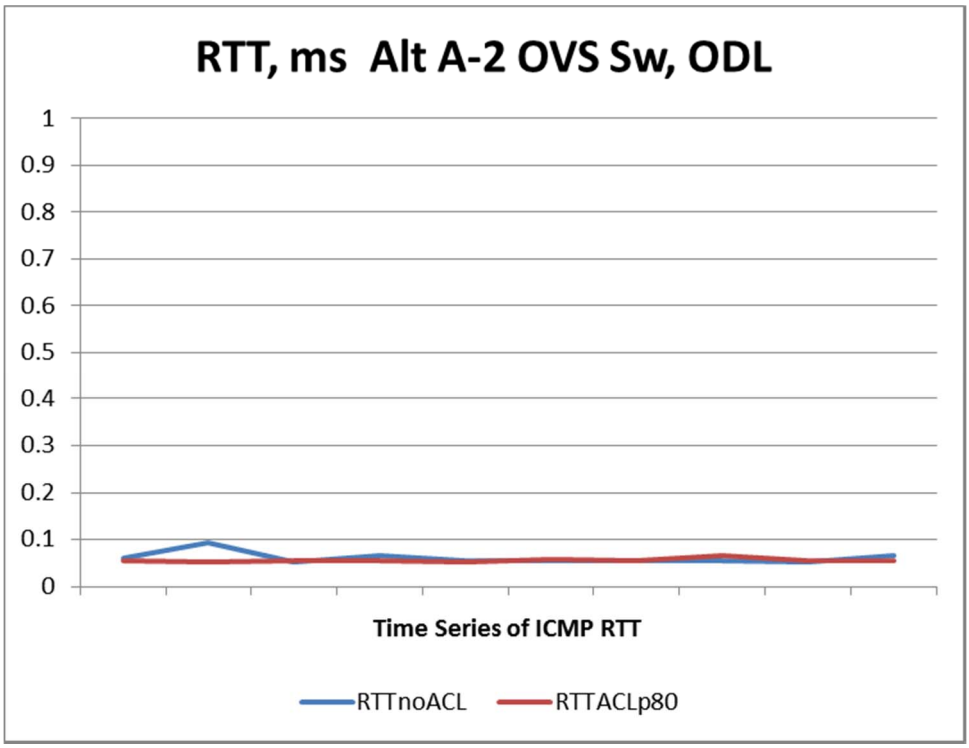


Figure B.4: Alternative A-2 OVS Switch, ODL Controller, ACLs to restrict Port 80

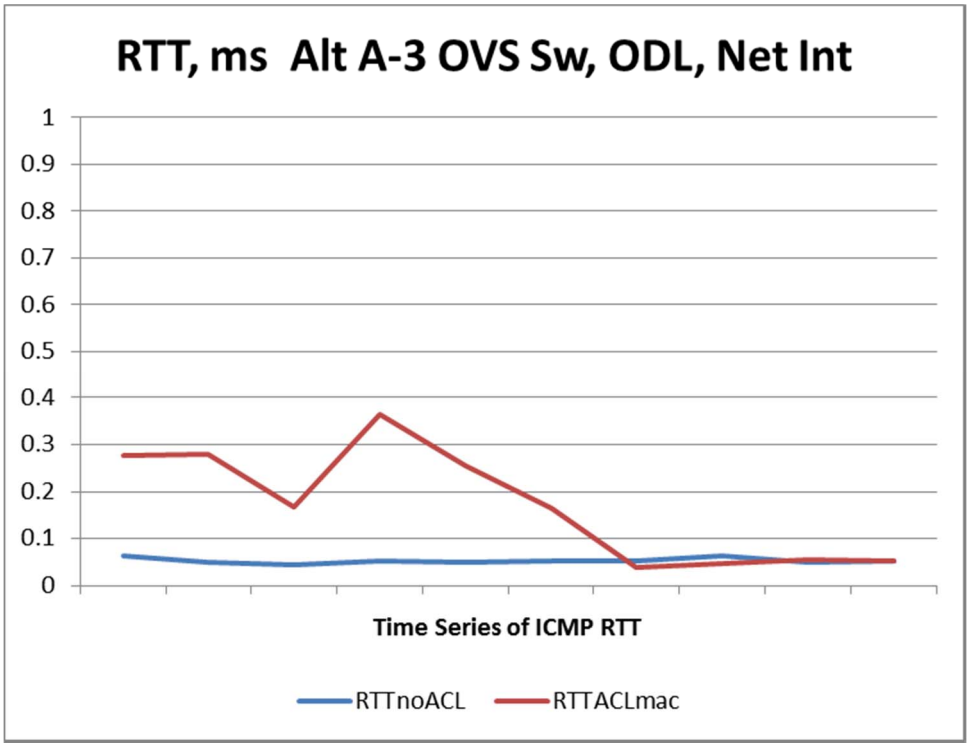


Figure B.5: Alternative A-3 OVS Switch, ODL, Network Intent to Block MAC combination

Annex C:
Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Mr. Al Morton, AT&T Labs

Other contributors:
Ms. Maryam Tahhan, Intel Corporation

Ms. Sylvia Almagia, ETSI CTI

Mr. Akram Al Sawaf, EANTC

Mr. Carsten Rossenhoevel, EANTC

Mr. Pierre Lynch, Ixia Technologies

Mr. Gergely Csatari, Nokia

Mr. Bruno Chatras, Orange

Mr. Joerg Aelken, Ericsson Telefonaktiebolaget LM

History

Document history		
V1.1.1	May 2017	Publication
V1.1.2	July 2017	Publication