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

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
DGS/NFV-TST001

Keywords
benchmarking, NFV, testing, validation

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

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.

Foreword

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

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**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.

1 Scope

The present document is an informative report on methods for pre-deployment testing of the functional components of an NFV environment. The NFV components addressed in the present document include Virtual Network Functions (VNFs), the NFV Infrastructure (NFVI) and the NFV Management and Orchestration (NFV MANO). The recommendations focus on lab testing and the following aspects of pre-deployment testing:

1) Assessing the performance of the NFVI and its ability to fulfil the performance and reliability requirements of the VNFs executing on the NFVI.

2) Data and control plane testing of VNFs and their interactions with the NFV Infrastructure and the NFV MANO.

3) Validating the performance, reliability and scaling capabilities of Network Services.

2 References

2.1 Normative 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.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

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 necessary for the application of the present document.

[1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

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-SWA 001: "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture".

[i.2] IETF RFC 2544: "Benchmarking Methodology for Network Interconnect Devices".

[i.3] IETF RFC 2889: "Benchmarking Methodology for LAN Switching Devices".

[i.4] IETF RFC 5180: "IPv6 Benchmarking Methodology for Network Interconnect Devices".

[i.5] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[i.6] ETSI GS NFV-INF 010: "Network Functions Virtualisation (NFV); Service Quality Metrics".

[i.7] ETSI GS NFV 001: "Network Functions Virtualisation (NFV); Use Cases".

[i.8] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.9] ETSI GS NFV-PER 001: "Network Functions Virtualisation (NFV); NFV Performance & Portability Best Practises".

[i.10] IETF draft-vsperf-bmwg-vswitch-opnfv-01: "Benchmarking virtual switches in OPNFV".

[i.11] IETF RFC 4656: "One Way Active Measurement Protocol".

[i.12] IETF RFC 5357: "Two Way Active Measurement Protocol".

[i.13] One-Way Active Measurement Protocol (OWAMP).

NOTE: Available at http://software.internet2.edu/owamp/.

[i.14] IETF draft-ietf-bmwg-virtual-net-01: "Considerations for Benchmarking Virtual Network Functions and Their Infrastructure".

[i.15] IETF draft-huang-bmwg-virtual-network-performance-01: "Benchmarking methodology for Virtualisation Network Performance".

[i.16] ETSI GS NFV-INF 004: "Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain".

[i.17] ETSI TS 123 002: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Network architecture (3GPP TS 23.002)".

[i.18] ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

[i.19] ETSI TS 122 278: "Universal Mobile Telecommunications System (UMTS); LTE; Service requirements for the Evolved Packet System (EPS) (3GPP TS 22.278)".

[i.20] IETF RFC 5481: "Packet Delay Variation Applicability Statement".

[i.21] IETF RFC 6985: "IMIX Genome".

[i.22] IETF RFC 2647: "Vocabulary for 3GPP Specifications".

[i.23] IETF RFC 3511: "Service Requirements for the Evolved Packet System (EPS)".

[i.24] IETF RFC 6349: "Packet Delay Variation Applicability Statement".

[i.25] IETF RFC 7230 to IETF RFC 7239: The family of IETF RFCs that specify HTTP/1.1.

[i.26] IETF RFC 4271: "A Border Gateway Protocol 4 (BGP-4)".

[i.27] IETF RFC 2328: "OSPF Version 2".

3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [1] and the following apply:

BFD Bidirectional Forwarding Detection
BGP Border Gateway Protocol
DoA Dead on Arrival
DUT Device Under Test
FUT Function Under Test
IMIX Internet MIX

NOTE: Some benchmarking methodologies use constant packet sizes, others use a mixture of packet sizes, or "IMIX" ("Internet Mix").

ISIS Intermediate System to Intermediate System
LDP Label Distribution Protocol
NSUT Network Service Under Test

OSPF	Open Shortest Path First
OWAMP	One Way Active Measurement Protocol
RSVP	Resource ReserVation Protocol
SUT	System Under Test
TWAMP	Two Way Active Measurement Protocol
VNFUT	Virtual Network Function Under Test
WG	Working Group

4 Definition of SUTs

4.1 Overview

All the recommended test methods (e.g. functional testing, performance testing etc.) address a certain target to be validated and a test environment enabling the test execution. A test target in the context of the present document is considered to be the System Under Test (SUT) which comprises one or more Functions Under Test (FUT).

The following clauses describe the general definitions of SUTs, the test environment, the test function and the NFV components considered as SUTs for pre-deployment validation.

All descriptions provide a functional view; connections between elements in the figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 illustrate functional interaction.

4.2 System Under Test (SUT)

In the context of pre-deployment validation, the System Under Test (SUT) consists of one or more functions under test.

NOTE: The functions under test (FUT) are entities which are also commonly known as Devices Under Test (DUT) in the testing community. The term Device Under Test is not used in the present document in order to avoid ambiguities; devices are often considered to be physical entities which does not apply here.

In order to illustrate this concept, the functions under test could for example be implementations of functional blocks from the NFV architecture such as virtualisation layer or VNF. However, other physical or virtual components could as well be functions under test (FUT), like a virtual switch for example.

Each test specification validates one SUT where the SUT is one or more functional components of the NFV architecture. The SUTs considered for pre-deployment validation are the NFV Infrastructure (NFVI), a Virtualised Network Function (VNF), a Network Service (NS) or the Management and Orchestration (MANO).

It has to be noted that even though the MANO or parts of it are listed as potential SUTs, no direct pre-deployment validation methodologies of them are in the scope of this report. However they are required as supporting functional blocks for the validation of other entities and are listed for completeness and might be considered for further study.

4.3 Test environment

The test environment for pre-deployment validation consists of reference implementations of those functional NFV components from the NFV architecture which do not represent the particular SUT. Additionally the test environment contains test functions and entities to enable controlling the test execution and collecting the test measurements.

4.4 Test function

The test functions for pre-deployment validation are entities that communicate with the SUT via standardized interfaces. The test functions are controlled from the test environment for test execution and are monitored from the test environment to obtain measurements for test results.

4.5 NFV Infrastructure Under Test

For pre-deployment validation of the NFV Infrastructure (NFVI), the NFVI represents the SUT.

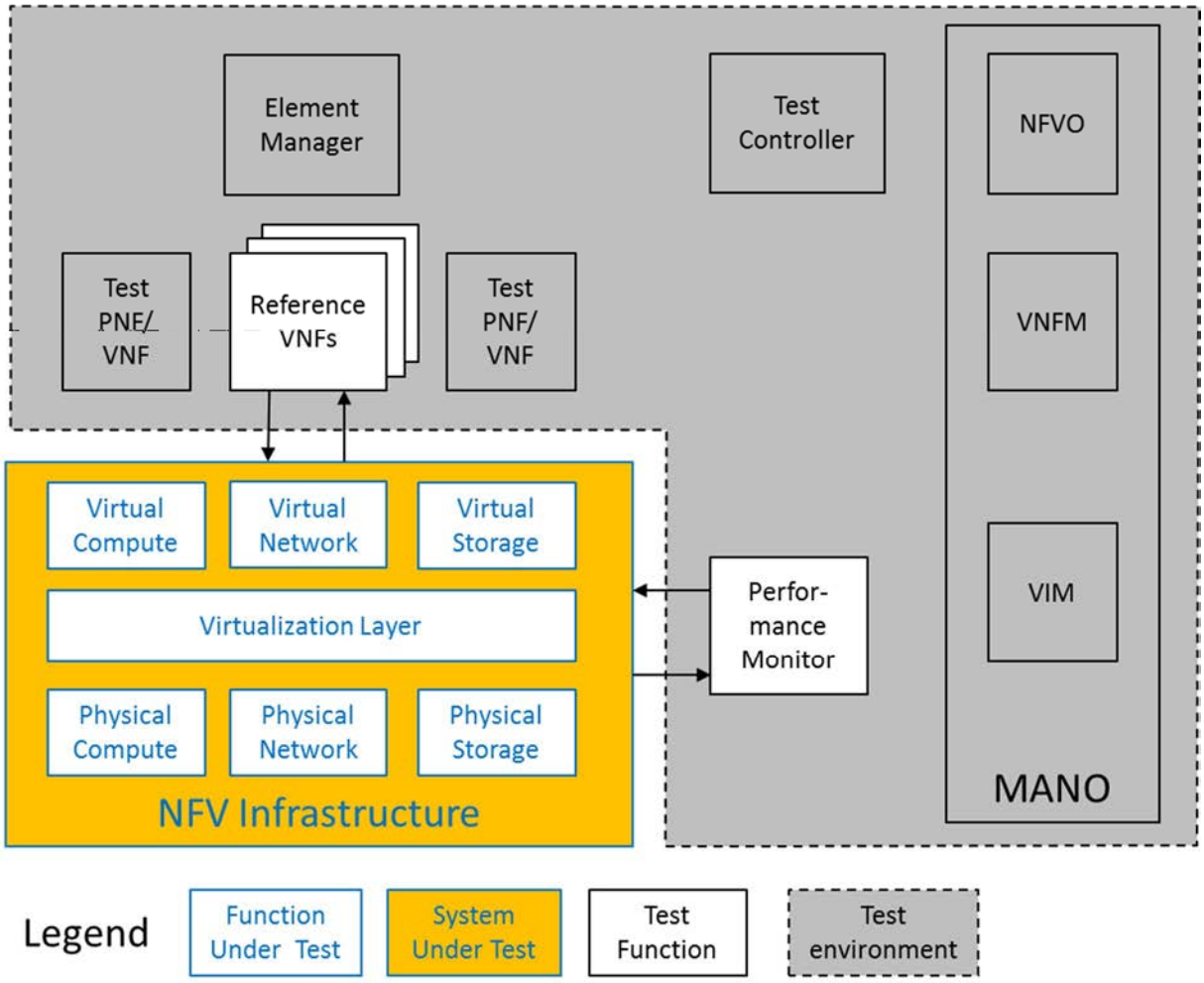

Figure 4.1: Functional architecture for NFVI under test

As illustrated in figure 4.1, the SUT comprises of the following functions under test (FUT):

- Physical Compute
- Physical Network
- Physical Storage
- Virtualisation Layer
- Virtual Compute
- Virtual Network
- Virtual Storage

The test environment consists of a reference implementation of the NFV MANO functional components plus a Test Controller, Test PNFs/VNFs, Reference VNFs and a Performance Monitor. In case required for maintaining the test and reference PNFs/VNFs, an optional Element Manager might be part of the test environment as well.

Different Reference VNFs as test functions are required to cover all aspects concerning different VNF types. The Reference VNFs are expected to be of the types described in ETSI GS NFV-SWA 001 [i.1], annex B, and shown in figure 4.2.

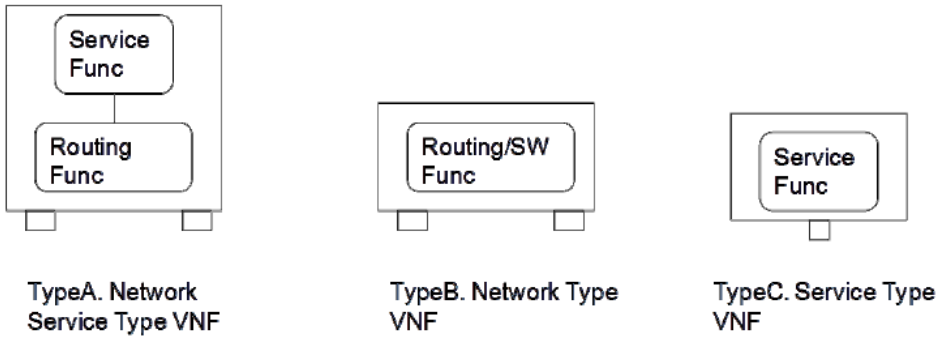

Figure 4.2: Reference VNF types (ETSI GS NFV-SWA 001 [i.1])

A Performance Monitor as test function is required to measure the performance indicators from the NFVI.

Optional test PNFs/VNFs might be required for certain test methods to enable traffic scenarios towards the Reference VNFs.

4.6 VNF Under Test

For pre-deployment validation of a Virtualised Network Function (VNF), the SUT consists of one FUT which is the VNF Under Test, see figure 4.3.

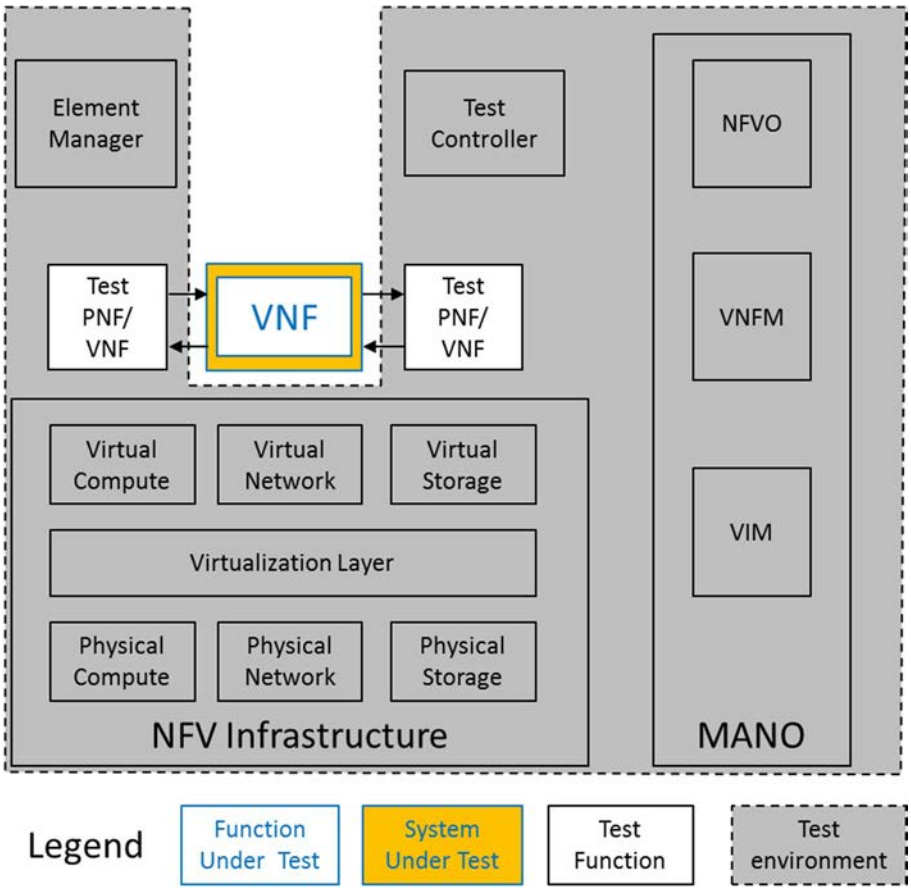

Figure 4.3: Functional architecture for VNF Under Test

The test environment consists of reference implementations of NFVI and NFV MANO functional components plus a Test Controller and Test PNFs/VNFs. In case required for maintaining the test PNFs/VNFs and the VNF Under Test, an optional Element Manager might be part of the test environment as well.

The Test PNFs/VNFs enable traffic scenarios towards the VNF Under Test and provide interfaces exposing access to functional and performance indicators.

4.7 NS Under Test

For pre-deployment validation of a Network Service (NS), the NS represents the SUT.

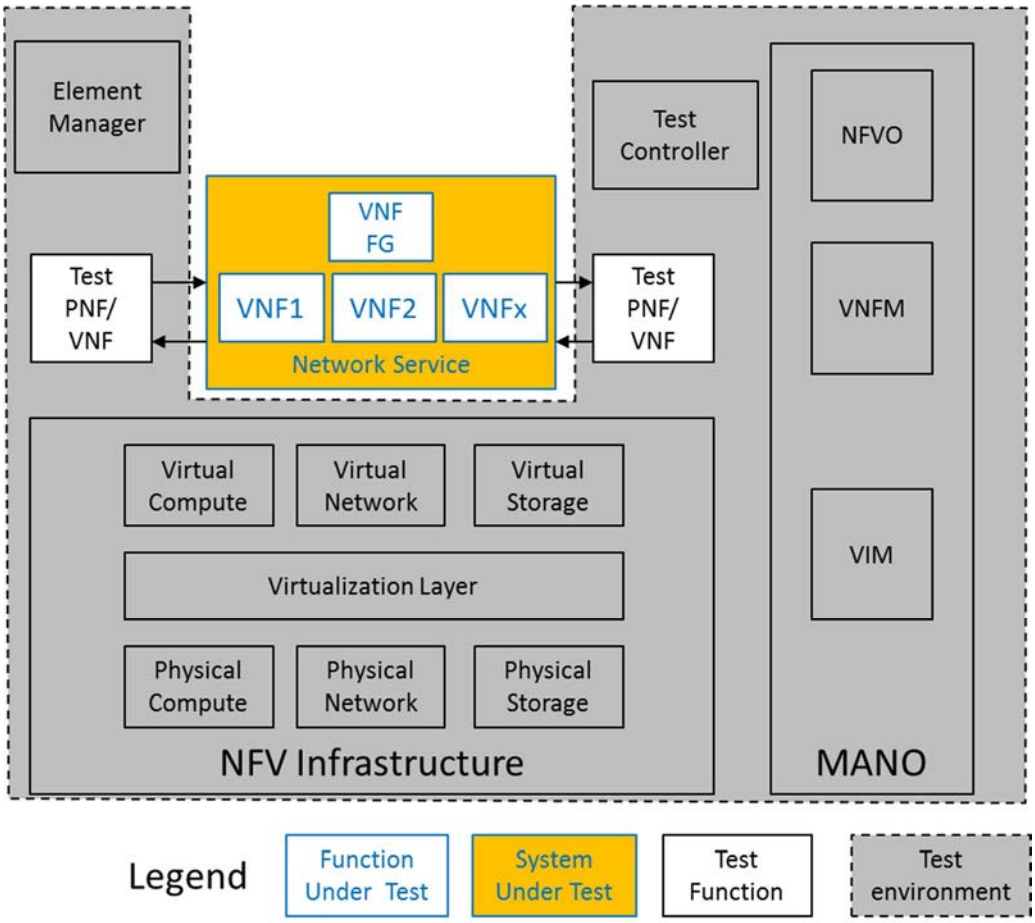

Figure 4.4: Functional architecture for NS Under Test

Note that in figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6, there is a physical overlap between the SUT and the NFVI in the Test Environment. For example, the VNF FG overlaps with the Virtual Network aspect of the NFVI.

The SUT consists of two or more VNFs and a VNF Forwarding Graph (VNF FG) which represent the Functions Under Test respectively.

The test environment consists of reference implementations of NFVI and NFV MANO functional components plus a Test Controller and Test PNFs/VNFs. In case required for maintaining the test PNFs/VNFs and the VNFs as FUTs of the NS Under Test, an optional Element Manager might be part of the test environment as well.

The Test PNFs/VNFs enable traffic scenarios towards the NS Under Test and provide interfaces exposing access to functional and performance indicators.

4.8 Management and Orchestration Under Test

For pre-deployment validation of the Management and Orchestration (MANO), the MANO represents the SUT. As mentioned before, no direct pre-deployment validation methodologies of the MANO are in the scope of the present document but the corresponding SUT is listed for completeness and for further studies.

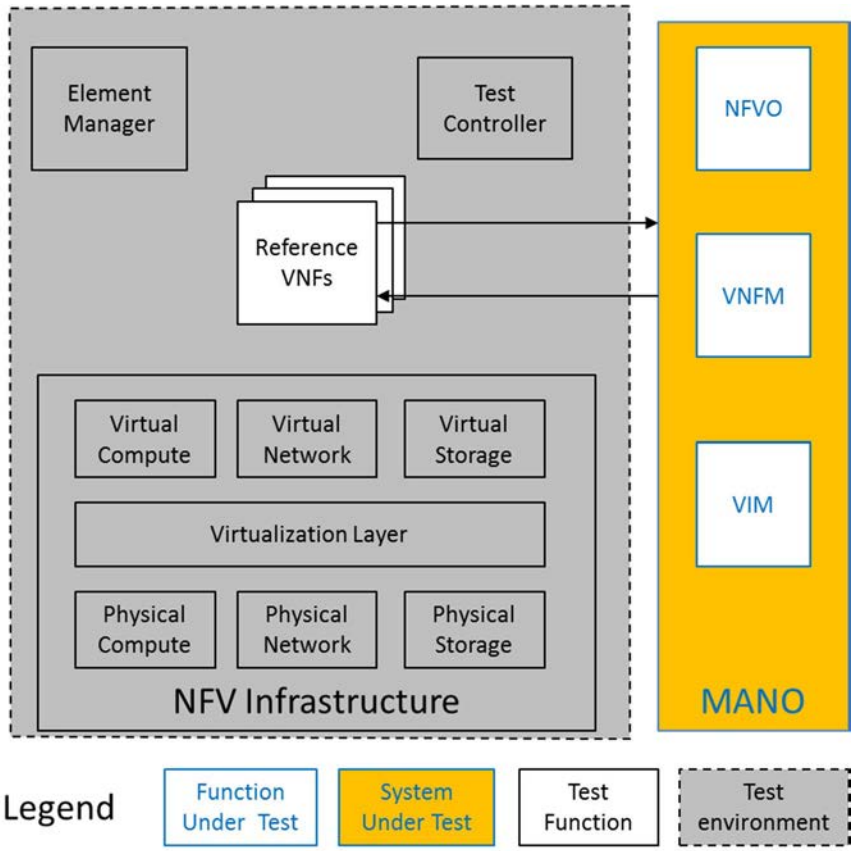

Figure 4.5: Functional architecture for MANO Under Test

The SUT consists of the NFV Orchestrator (NFVO), the VNF Manager (VNFM) and the Virtual Infrastructure Manager (VIM) which represent the functions under test respectively. See also figure 4.5.

The test environment consists of a reference implementation of NFVI plus a Test Controller and reference VNFs. In case required for maintaining the reference VNFs, an optional Element Manager might be part of the test environment as well.

Different Reference VNFs are required as test functions to cover all aspects concerning different VNF types. The Reference VNFs are expected to be of the types as described in ETSI GS NFV-SWA 001 [i.1], annex B, and shown in figure 4.2.

4.9 NFV Infrastructure + VIM Under Test

A variant of the NFVI Under Test could be acombination of the NFVI and the Virtual Infrastructure Manager (VIM) Under Test. For pre-deployment validation of the NFV Infrastructure (NFVI) including the VIM, the NFVI and the VIM represent the SUT. Even though this report does not contain direct pre-deployment validation methodologies for this combination, it is listed for completeness and for further studies.

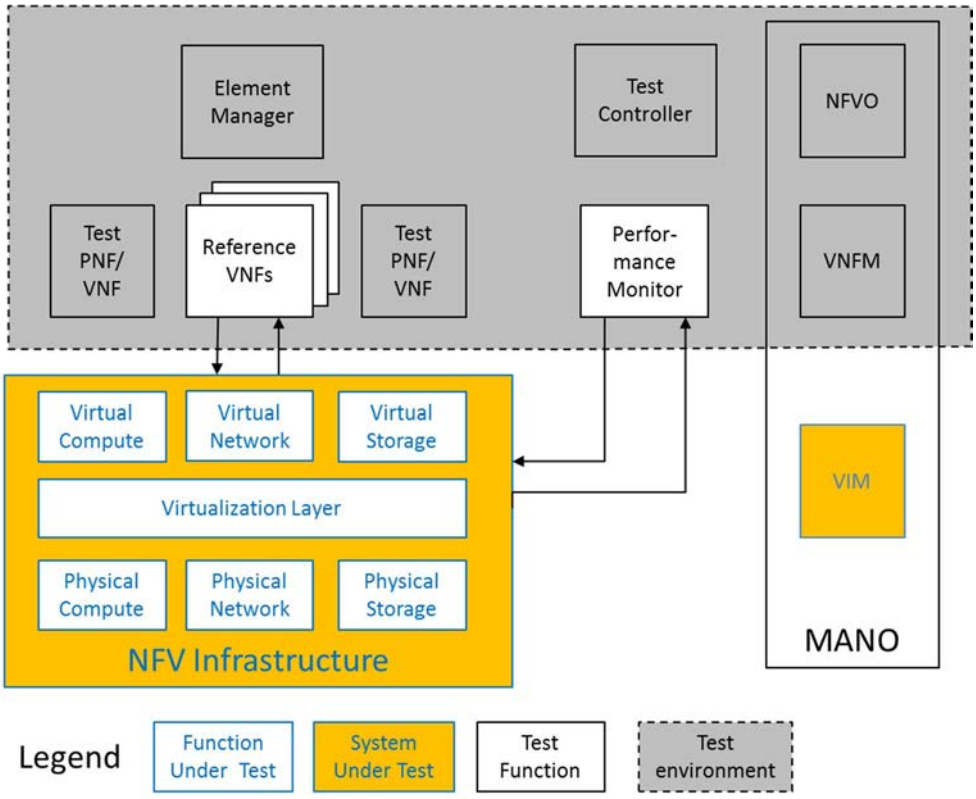

Figure 4.6: Functional architecture for NFVI + VIM Under Test

As illustrated in figure 4.6, the SUT comprises of the following functions under test (FUT):

- Physical Compute
- Physical Network
- Physical Storage
- Virtualisation Layer
- Virtual Compute
- Virtual Network
- Virtual Storage
- Virtual Infrastructure Manager

The test environment consists of a reference implementation of the NFV MANO functional components excluding the VIM plus a Test Controller, Test PNFs/VNFs, Reference VNFs and a Performance Monitor. In case required for maintaining the test and reference PNFs/VNFs, an optional Element Manager might be part of the test environment as well.

Different Reference VNFs as test functions are required to cover all aspects concerning different VNF types. The Reference VNFs are expected to be of the types described in ETSI GS NFV-SWA 001 [i.1], annex B, and shown in figure 4.2.

A Performance Monitor as test function is required to measure the performance indicators from the NFVI.

Optional test PNFs/VNFs might be required for certain test methods to enable traffic scenarios towards the Reference VNFs.

5 Test methods for pre-deployment validation of SUTs

5.1 Validating physical DUTs and SUTs

5.1.1 Overview

This clause provides a high level description of the methods used to validate physical DUTs and SUTs (e.g. individual or network of purpose built routers, switches and appliances) prior to their deployment in the field. Its purpose is to help the reader understand the differences between the testing methods employed in physical and virtual/NFV environments.

Physical DUTs are traditionally validated using physical 'test devices'. The test device interoperates with the DUT in a lab setup. The test device establishes sessions with the DUT and exchanges user plane and control plane traffic to assess the functionality and performance of the DUT. Three representative use cases for validation of physical DUTs are presented in clauses 5.1.2, 5.1.3 and 5.1.4.

5.1.2 Data plane validation

Standards based benchmarking methods are used to perform data plane validation of the physical DUT(s). A few of the most significant benchmarking methods are listed below:

- IETF RFC 2544 [i.2] specifies methods to assess network interconnect devices and measures metrics such as throughput, latency, frame loss rate, and system recovery time.
- IETF RFC 2889 [i.3] specifies methods for benchmarking of LAN switching devices and takes into consideration flooding and MAC address learning.
- IETF RFC 5180 [i.4] extends IETF RFC 2544 [i.2] for IPv6 capable DUTs and networks.

In these benchmarking methods, a test device originates test traffic. The traffic is received, processed and forwarded by the DUT(s) and terminated on another test device. The originating test device varies the frame sizes, burst sizes and frame rates and the terminating test device measures metrics such as throughput, latency and frame loss rates. The DUTs are connected to the test devices as shown in figure5.1. Each of the test devices can be physically connected to the DUT on multiple (perhaps hundreds) of ports using a variety of speeds (1G, 10G, 40G, 100G, etc.) and a variety of interface types (Ethernet, ATM, Fibre channel, etc.). Please refer to the IETF RFC 2544 [i.2], IETF RFC 2889 [i.3] and IETF RFC 5180 [i.4] for a detailed explanation of the testing methods.

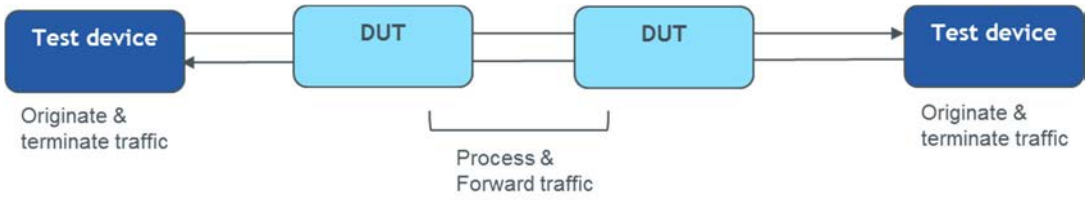

Figure 5.1: Test setup for data plane benchmarking of physical DUT(s)

5.1.3 Control plane benchmarking

The testing of a DUT for compliance to IETF RFC based protocols (for example IETF RFC 4271 [i.26] for BGP and IETF RFC 2328 [i.27] for OSPFv2) is accomplished by connecting it to test devices that speak the same control plane protocols. In figure 5.2, Test device 1 emulates a network of nodes that speak the same protocols as the DUT. Test device 1 establishes control sessions (for e.g. BGP, OSPF, ISIS, RSVP, LDP and/or BFD) with the DUT, exchanges routes, and defines traffic flows that are bound to these sessions. In effect, Test device 1 exchanges both control and data plane traffic with the DUT. The DUT forwards the traffic which then terminates on Test device 2.

The Test devices benchmark the DUT by using the following methods:

- Scale up the number of control sessions between the test device and DUT to benchmark the maximum session scale supported by DUT.
- Vary the session set up and tear down rates to assess DUT performance.

- Verify that the DUT layers the control plane packets in the proper order (e.g. VPN traffic exchanged using correct underlying label stack, DHCP over VPLS, etc.).
- Test device 2 originates data traffic destined for addresses advertised by Test device 1's routing sessions with DUT. The ability of the DUT to correctly forward traffic from Test device 1 toward Test device 2 is validated.

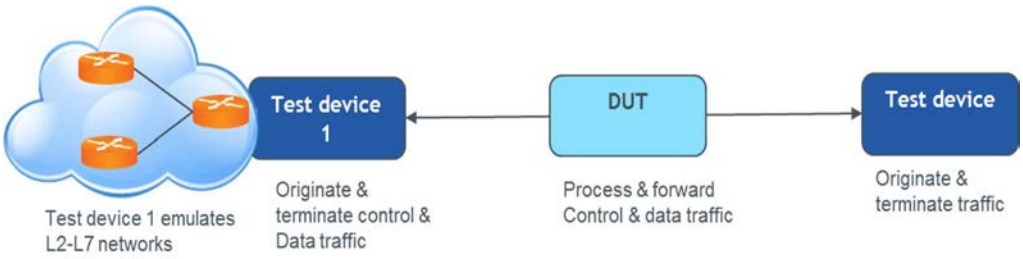

Figure 5.2: Test setup for control plane benchmarking of physical DUT(s)

5.1.4 Management plane validation - Testing fault detection, recovery and convergence

The fault detection, recovery, and convergence capabilities of the DUT are validated by connecting 3 Test device ports to the DUT as shown in figure 5.3. Test devices 2 and 3 will advertise identical routes to a destination A, with Test device 2 advertising a lower cost route. All traffic originated by Test device 1 for destination A will be forwarded by the DUT to Test device 2 (which advertised lower cost route). This methodology is applicable for a wide range of routing protocols that are used to exchange routes between the DUT and test devices.

- Test device 2 injects an error, such as withdraw route, break link, or BFD Stop.
- The test devices assess the DUT's ability to a) detect the fault quickly, b) install the backup route, c) stop traffic toward Test device 2 and forward affected traffic toward Test device 3.
- The Test devices will work in synchronization to measure the fault detection and convergence times with microsecond accuracy.

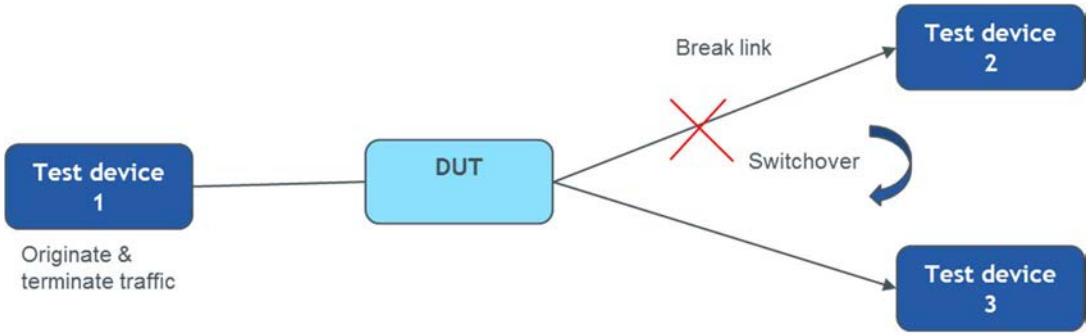

Figure 5.3: Test setup for management plane testing of physical DUT(s)

5.2 Impact of virtualisation on testing methods

To understand the impact of virtualisation on testing methods, it is instructive to revisit the NFV framework defined in ETSI GS NFV 002 [i.5]. The physical DUTs described in the clause 5.1 are instantiated and executed as VNFs in an NFV environment. In addition, NFV architecture defines new entities such as the NFVI and the NFV MANO and new interfaces between the VNFs, NFVI and the NFV MANO components.

The new components and the new interfaces defined by the NFV architecture introduce new failure points and mandate the need for additional testing methods to ensure the reliability of VNFs and services.

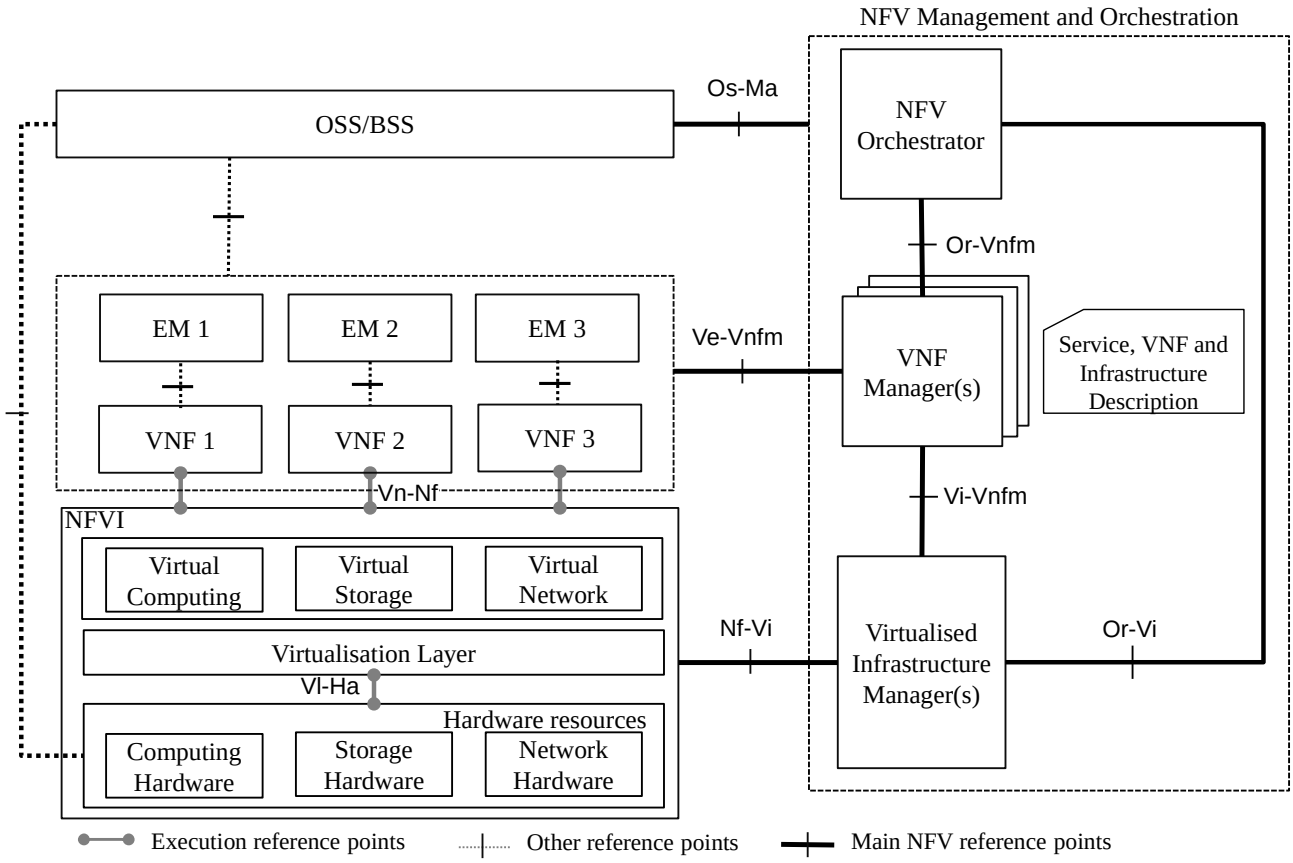

Figure 5.4: The NFV architectural framework with reference points

New capabilities introduced by virtualisation that change the way systems are tested are:

- In traditional networks, NFs run on dedicated physical devices with dedicated resources. In a virtualised environment, VNFs run in a shared compute, storage and network environment and may contend for the same resources.
- The Virtualisation Layer, consisting of a hypervisor, OS container and/or vSwitch, abstracts the resource details from the VNFs. The performance of the NFV Infrastructure is influenced by the type of load (network versus IT workload, CPU intensive, memory intensive or storage intensive) and number of VNFs executing.
- Special data plane acceleration mechanisms may be used for IO intensive applications. Examples of such mechanisms are DPDK and SR-IOV, which allow VNFs to bypass bottlenecks in the Virtualisation Layer while transmitting and receiving packets.
- NFV allows for service chaining, where a forwarding graph may be designed to define the path a packet flow will take from its source to its destination. The path may consist of one or multiple VNFs, which may or may not be present on the same NFVI.
- A VNF will be instantiated with a defined amount of resources available to it. However, NFV allows for the MANO function to dynamically modify the amount of resources allocated to a VNF, as well as instantiate other VNFs, as the load requires.
- Failure recovery mechanisms allow for a VNF to be transferred to another NFVI or another VNF instantiated to recover from a catastrophic error.

These new capabilities of NFV warrant new methods of testing and monitoring the network. Therefore, test plans will be constructed to validate these new capabilities. While these concepts are largely independent of the function a VNF accomplishes, the tests will take into account the type of function the VNF executes in order to drive the appropriate traffic type for a test. For example, a Diameter server VNF will require different traffic input and output than a firewall VNF. The methods described below will focus on the new capabilities introduced by NFV, while providing examples of traffic types. However, the focus is not on the VNFs' specific functionality.

The main variables (configuration items) in virtualised system are listed below:

- Resources allocated to the VNF: compute cores, memory allocation, etc.
- Resources allocated to the vSwitch.
- Virtualisation layer (hypervisor) used.
- The HW resources, including compute, networking (NIC) and storage used.
- The usage (or not) of data plane acceleration techniques.
- The MANO policy for scaling a VNFs resources or to instantiate new VNFs to handle increased load. The opposite is true when the load drops and the VNFs can be scaled back.
- The presence or absence of other VNFs on the same NFVI (multi-tenancy), and the function of these VNFs.

Most of the tests below will exercise the concepts above, while having either fixed or variable values for the configuration values. Depending on the objective of the test, some configuration values will be fixed, some may vary per iteration of the test, and others will be measured, and will become the result of the test. The tests will also have as an objective to discover the optimal settings of these configuration variables, for the desired performance of the system, thus helping in dimensioning the system appropriately.

5.3 Common test methods and specifications for virtual environments

There are multiple reasons to perform pre-deployment testing, most of which are not new to NFV:

- Feature verification.
- Regression testing when SW or HW changes are made.
- Availability and robustness verification.

However, some new concepts are introduced when performance testing is concerned. This is because the very nature of virtualisation introduces many new variables and controls that affect performance, as listed above (multi-tenancy, acceleration techniques, etc.). With this in mind, it leads to different means of approaching the following broad categories of performance testing. These categories are not all-inclusive, but they include the majority of performance testing. It is important to note that the discussion that follows applies to all types of pre-deployment testing, and not only to performance testing. Performance testing is used as an example to illustrate the concepts.

1) Performance verification: The goal is to validate that a set of established numerical performance objectives can be reached, under mostly fixed conditions within the SUT. This is typically done to verify that the published performance metrics for a SUT (for example a VNF or an NFVI platform) can be met.

2) Benchmarking: This type of test is aimed at finding out the maximum performance level of a SUT with fixed resources, conducted within an isolated test environment (ITE). It is a goal seeking test exercise, where iterations of the test are run successively, in order to reach the maximum of the performance metric of interest.

3) Dimensioning: This type of test aims to determine the amount of infrastructure required to support a defined set of performance metrics. The performance metrics are known, the objective is to determine the amount of NFVI resources required to support the performance levels.

These are not really new categories of testing, but what is new to NFV is how to go about the testing. A widely adopted strategy for performance testing of a SUT is to isolate the SUT in order to reduce the amount of variables in the test. This makes it easier to ensure that the performance being measured is that of the SUT itself, without being influenced by other devices, and it also makes it easier to repeat deterministic configurations and results. With dedicated HW platforms supporting today's networking devices, it is possible to isolate the SUT effectively, and to remove all other variables from a performance test. An example of this (from mobility architecture) is the ability to isolate an S/PGW for performance testing by simulating the surrounding elements with test devices (the MME, the eNodeB, the PDN elements, etc.)

However, by the nature of the NFV architecture, it is very challenging to isolate one function as a SUT without the presence of the other functions supporting the SUT. For example, it is not possible to test a specific VNF (as a SUT) without having the NFVI present. Since the NFVI seriously impacts the performance of the VNF, this presents a challenge for performance testing the VNF. In fact, in a typical NFV deployment, other VNFs can be running on the same NFVI (multi-tenancy), which further complicates the testing.

The recommended way to approximate SUT isolation in an NFV environment is to strictly control the configuration parameters of the NFV architecture elements that do not comprise the SUT (i.e. the test environment) while varying configuration parameters for the SUT. This leads to two different sets of configuration parameters:

1) The fixed configuration parameters: these parameters remain constant for all iterations of a test exercise.

2) The variable configuration parameters: these can be modified between iterations of a test exercise.

The categories of performance tests above then help to define, from the total set of configuration parameters, which fall into the fixed and variable parameters. The definition of the fixed and variable configuration parameters determine the components that are isolated for the performance test (i.e. to isolate the SUT as much as feasible) and the test environment. It should be noted that variable configuration parameters are only modified between test run iterations.

EXAMPLE 1: Performance verification of a VNF:

Typically, the supplier of a VNF will have a performance guarantee for the VNF under strict conditions. The numerical guarantees are based on performance metrics such as packets/sec, throughput, etc. while the strict conditions will define, among other parameters, the NFVI configuration under which the performance metrics can be guaranteed. Thus, the configuration parameters for the NFVI platform will become part of the fixed set: Server brand and model, CPU, assigned cores, memory allocation, virtualisation layer, vSwitch and its configuration, etc. The variable configuration parameters will largely become configuration of the VNF itself.

EXAMPLE 2: Benchmarking a VNF:

The goal in this example is to discover the maximum performance level attainable for a VNF, given a fixed pool of resources provided by the NFVI. The exercise can often also involve optimization of the platform for the particular needs of the VNF under test. In this case, the SUT technically could also include the NFVI itself. Therefore, the sets of configuration parameters (non-exhaustive) would look like this:

- Fixed: HW resource amounts (servers, cards, CPUs, memory, vSwitch, hypervisor, etc.).
- Variable: Core and memory allocation, CPU pinning, acceleration techniques used (or not), etc.

This is a goal seeking type of test, meaning that iterations of the test are run, changing the variables between iterations, in order to achieve the maximum performance. It is recommended practice to change the minimum amount of variable parameters for each iteration, in order to understand the impact of each variable parameter individually on performance, and the interactions between parameters.

EXAMPLE 3: Dimensioning for a VNF:

For this example, the same VNF gets subjected to performance testing, but the NFVI may be different from that specified by the VNF supplier. It may be NFVI already selected by the platform supplier. In this case, the test objective will be to discover the amount of NFVI resources that are required to support a specified set of performance levels (which could be expressed by metrics such as latency, throughput or packets/sec, etc.)

- Fixed: Core and memory allocation per VM, CPU pinning, acceleration techniques, vSwitch and its configuration.
- Variable: HW resources.

This is also a goal seeking test, where the performance metrics to be fixed are known, and the goal is to determine the amount of resources to fulfil these metrics. It may also be extended to include other VNFs running simultaneously in order to introduce real-world deployment concepts like multi-tenancy.

There are many resources that have exhaustive lists of the configuration parameters of the NFVI and MANO policies that impact performance: please refer to ETSI GS NFV 002 [i.5], ETSI GS NFV-INF 010 [i.6] and ETSI GS NFV 001 [i.7] for details.

5.4 Considerations on choice of virtualised versus hardware based test appliances

Just like the network functions themselves, the test devices can be virtualised as well. Traditional hardware-based tools now have virtual versions that can also produce the same test inputs and outputs. This leads to a decision on what type of test devices to use: virtual or physical.

There are no rules to select one or the other, but rather a series of considerations that can guide the selection.

- Development testing: for this type of testing, where the stress levels are low and a very interactive type of testing is the goal, then the convenience and ease of being able to launch and distribute a virtual test device are high.

- Test device contention: if many users are accessing the test devices simultaneously, virtual test tools have an advantage over hardware based tools. Whenever a test device is needed, simply launching the virtual tool and not having to worry about someone else using it is a definite advantage.

- Geographically dispersed test platforms: if multiple testing environments are located in different geographical areas, having virtualised test devices is easier than having to equip the platforms with different hardware.

- Test device orchestration: modern test devices have automation APIs, which may be sufficient. However, when a goal is to have the test device being orchestrated along with the rest of the NFV environment, then this leads to the necessity of having a virtualised test device with open and standard APIs that support the NFV platform.

- Performance testing: both virtualised and physical test devices have advantages in this case.

    - Virtualised test devices have the advantages stated above, and with a sufficiently equipped NFVI platform, can reach very high levels of performance. A mechanism should be used in order to not compete with resources with the system under test. This is important: the test device should not compete for the same resources as the SUT (i.e. in a multi-tenant situation), or otherwise impact the performance of the SUT, else the performance test results will not be reliable.

    - Physical test devices have the advantage of having known performance expectations for stress testing, such that it removes a variable from the testing equation. Also, the test device will not impact the SUT performance in any way.

- Measurement accuracy: if very precise measurements are required on the traffic, then physical, HW-based test devices have an advantage. They are better equipped for precise time-stamping of the incoming and outgoing packets than today's virtual solutions. It is recommended that users perform baseline tests of useful clock precision and clock stability for both physical and virtual test devices.

- The system under test (SUT) and its access/egress interfaces: The SUT definition itself can direct the choice of test device. If one test objective is for east-west traffic between VNFs on the same platform, then virtualised test devices are appropriate. However, if the NFVI or its components are part of the test, then either a physical test device or a virtualised test device (that does not compete with the SUT resources) can be recommended. The access and egress interfaces of the SUT may determine whether the test device is physical or virtual or a combination of both types. For example, the east-west traffic between VNFs may only be accessible on virtual interfaces. SUT interface types are expected to have a profound influence on the measured results.

- Deployment environment testing: if the test device will not only conduct pre-deployment testing, but would also be shipped with the platform such that it can also run tests on the active, deployed network, then a virtualised test device is an obvious choice.

6 Pre-deployment validation of NFV Infrastructure

6.1 Introduction

Telecom networks are realized by connecting diverse applications and functions to achieve the required overall functionality. Applications and functions which are part of a telecom network require high reliability, thus should be dimensioned with predictable performance. Applying virtualisation to those applications or Network Functions, e.g. the use cases defined in ETSI GS NFV 001 [i.7], impacts the infrastructure owner, the application owner and the VNF application vendor. From an application owner perspective, this translates to a validation target that imposes the total system downtime for a NFV infrastructure to be minimized, including application as well as infrastructure maintenance activities, such as:

- Application/Infrastructure software faults.
- Application/ Infrastructure configuration faults.
- HW faults.
- HW repair/HW extension.
- Software/ Infrastructure upgrade.

Hence, the impact on the VNF application from disturbances in the infrastructure, e.g. infrastructure upgrade, as well as application software faults, should be identified and consequently minimized. In order to verify those stringent requirements when running VNF applications on a NFV infrastructure, extensive and complex end-to-end testing involving the infrastructure, the application and the network should be performed, resulting often in faults/bottlenecks which are complex and time-consuming to understand, correct and verify. It is desirable to find faults at an earlier stage, by utilizing simple test cases that examine basic infrastructure metrics.

This clause describes the methodology to validate the NFV infrastructure (NFVI) by using simple test cases to avoid basic infrastructure faults/bottlenecks being discovered late in the end to end validation process of the VNF applications. A potential fault, system limit or bottleneck is defined as a metric which needs to be validated by a test program. In the event of a basic NFV infrastructure fault/bottleneck appearing in the end to end verification, a root cause analysis should be triggered to improve the methodology so the fault/bottleneck could be detected by a new metric or combinations of metrics. The golden rule for the NFV infrastructure validation is: find as many faults/bottlenecks as early as possible, using simple test case or combinations of test cases. Faults/bottlenecks discovered and corrected in the early NFV infrastructure validation phase significantly reduce the total VNF application verification cost.

The methodology for pre-deployment validation of NFV infrastructure consists of identifying the type of the VNF application, breaking down the VNF type requirements into metrics where each metric is represented by a test case, validated stand alone or grouped together. The methodology described can also be used to compare the validation results of different NFVI implementations or alternative configurations. The validation of the NFV infrastructure should take place prior to deploying VNF applications; the end to end characterization requires the subsequent integration and validation of the VNF application which is not in the scope of this methodology.

The methodology for pre-deployment validation of the NFV infrastructure provides input to understand critical aspects such as:

- Is the NFV infrastructure under test suitable for the target VNF application? Does the NFV infrastructure fulfil the basic VNF type requirements?
- How deterministic are the characteristics of the infrastructure and how should the VNF application be deployed to guarantee predictable performance?
- Is the NFV infrastructure able to handle hardware faults without disturbing running VNF applications?
- Are there NFV infrastructure bottlenecks triggered by VNF applications?

Each VNF application is unique and requires its own set of metrics to investigate whether the NFV infrastructure under test is able to fulfil the requirements from the VNF application. This methodology proposes recommendations for a common, vendor-independent, test framework which will provide:

- Support to configure the test environment used for the validation of the metrics. This means, number of virtual machines, amount of memory per virtual machine, amount of virtual cores per virtual machine, network configuration.
- Support for independent NFV infrastructure deployment. This means creating compute, networking and storage resources in the virtualisation layer using automated templates, including deployment rules such as affinity and anti- affinity for the virtual machines on host machines.
- Support to configure the test functions used for the metric under validation.
- Support to evaluate if QoS requirements are fulfilled.
- Data collection of the test results.

The common test framework is an enabler for the unified methodology for validating metrics on any NFV infrastructure.

6.2 Infrastructure characteristics

The methodology for pre-deployment validation of the NFV infrastructure is based on characteristics which are in the scope of responsibility of the infrastructure owner. As an example, the number of sent and received SIP messages between two virtual machines is not considered an infrastructure characteristic, due to the fact that the owner of the infrastructure cannot impact the SIP implementation used in the VNF application. On the other hand, the number of Ethernet frames caused by the size of SIP messages is an infrastructure characteristic, since the owner of the infrastructure provides Layer 2 connectivity between virtual machines.

The infrastructure characteristics also depend on what services the infrastructure owner provides. Layer 3 networking could either be handled by soft router provided by the VNF application vendor or implemented by the infrastructure owner as an infrastructure service. The pre-deployment validation of the NFV infrastructure does not define in detail what these infrastructure characteristics are as this depends on what the infrastructure owner provides.

To structure the development of test cases to be used for measuring how the NFV infrastructure characteristics impact the VNF application, the metrics used for infrastructure validation are divided into compute, storage and networking sub groups. Each sub group is organized in the following categories:

- Performance/Speed: Infrastructure characteristics used to understand VNF application performance when deployed on the NFV infrastructure under test. Processing speed (instructions per second) is an example of a compute performance metric.
- Capacity/Scale: NFV infrastructure characteristics used to understand the (maximum) capacity the VNF application is able to reach when deployed on the NFV infrastructure under test. The maximum throughput of Ethernet frames per second switched in the infrastructure is an example of a networking capacity metric.
- Reliability/Availability: NFV infrastructure characteristics used to understand the reliability and availability of infrastructure components provided to the deployed VNF. The disk mean-time-to-failure is an example of a storage reliability metric.

Table 6.1 lists the infrastructure metrics per sub-group and category.

Table 6.1: Infrastructure metrics per sub-group and category

	Performance/Speed	Capacity/Scale	Reliability/Availability
Compute	• Latency for random memory access • Latency for cache read/write operations • Processing speed (instructions per second) • Throughput for random memory access (bytes per second)	• Number of cores and threads • Available memory size • Cache size • Processor utilization (max, average, standard deviation) • Memory utilization (max, average, standard deviation) • Cache utilization (max, average, standard deviation)	• Processor availability (Error free processing time) • Memory availability (Error free memory time) • Processor mean-time-to-failure • Memory mean-time-to-failure • Number of processing faults per second
Network	• Throughput per NFVI node (frames/byte per second) • Throughput provided to a VM (frames/byte per second) • Latency per traffic flow • Latency between VMs • Latency between NFVI nodes • Packet delay variation (jitter) between VMs • Packet delay variation (jitter) between NFVI nodes	• Number of connections • Number of frames sent/received • Maximum throughput between VMs (frames/byte per second) • Maximum throughput between NFVI nodes (frames/byte per second) • Network utilization (max, average, standard deviation) • Number of traffic flows	• NIC availability (Error free connection time) • Link availability (Error free transmission time) • NIC mean-time-to-failure • Network timeout duration due to link failure • Frame loss rate
Storage	• Sequential read/write IOPS • Random read/write IOPS • Latency for storage read/write operations • Throughput for storage read/write operations	• Storage/Disk size • Capacity allocation (block-based, object-based) • Block size • Maximum sequential read/write IOPS • Maximum random read/write IOPS • Disk utilization (max, average, standard deviation)	• Disk availability (Error free disk access time) • Disk mean-time-to-failure • Number of failed storage read/write operations per second

6.3 Scenario validation

The methodology aims at validating the NFV infrastructure from the perspective of a VNF application. It identifies the VNF type of the VNF application and decomposes the VNF type typical requirements into a set of metrics which are in the responsibility of the infrastructure owner. The metrics are forming a metrics vector for one scenario validation and are validated by individual test cases, executed by the supporting test environment.

The test scenario, comprising of the SUT and the metrics vector, is the entity to be executed by the test environment for one scenario validation. The metrics vector defines the individual test cases to be selected for the scenario validation.

System Under Test

The System Under Test (SUT) is the NFV infrastructure, comprised by the following functions under test:

- Compute - computational resources, such as CPU, NIC, memory, caches.
- Storage - storage resources, such as disk.
- Networking - connectivity services, such as switching and routing.

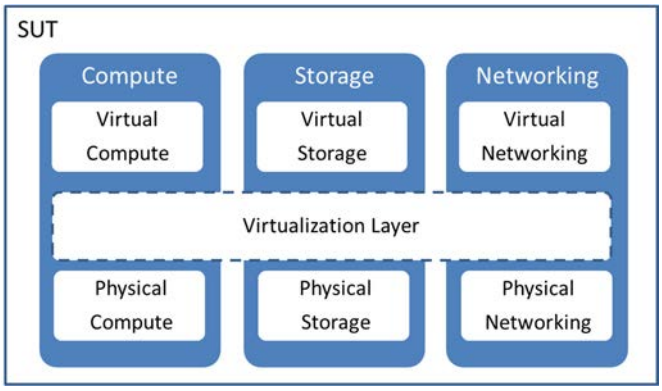

Figure 6.1: System Under Test for NFVI validation

Test Environment

The supporting test environment executes the following tasks as depicted in figure 6.2:

- Configure - Number of virtual machines acting as test functions (hosting the benchmarks/tools to execute the applicable metric validation), amount of memory per virtual machine, amount of virtual cores per virtual machine, network configuration, deployment rules such as affinity and anti-affinity.
- Deploy - Allocate compute, networking and storage resources in the virtualisation layer using automated templates, auto-generated or external. The test environment should also deploy appropriate workloads on the NFVI under test in parallel to the virtual machines acting as test functions to emulate key VNF characteristics.
- Test - Execute test cases to validate the metrics vector and collect results. The test cases use benchmarks/tools which are executed by physical or virtual test functions. The benchmarks/tools are supported by the test environment. The test includes definitions of criteria for pass/fail if a corresponding SLA value exists or includes definitions of benchmark results. The test environment triggers external scripts such as traffic generators or additional workloads to inject required SUT load or disturbances.
- Validate - Result presentation, pass/fail criteria if applicable.

Workload

In order to validate the requirements of a VNF application in terms of measuring relevant metrics it is important to consider VNF application typical workloads being deployed on the NFVI under test. The appropriate workloads could be provided by combinations of different means. They could be provided by VNFs implementing the VNF application under consideration, they could be provided by external traffic generators or they could be provided by the benchmarking/tools for metric measurements themselves. In case the benchmarking/tools are not performing passive measurements but inducing workloads themselves it has to be ensured that no unwanted interference with other workloads are created. The workloads are dynamically controlled by the test environment during scenario validation to match the requirements for the individual test case. Which workloads to be deployed and combined depends on the individual test scenario.

In all cases it is recommended to document the methods and parameters of the workload generation.

In order to understand the intricacies of the infrastructure-application interaction, such as whether an NFV infrastructure HW/SW upgrade is performed without disturbing running VNF applications, external stimuli to the system under test is required. The generic environment supports the use of external scripts to simulate disturbances, e.g. remove a compute blade to simulate HW faults, remove a switch, order live migration, insert a noisy neighbour, use an external traffic generator to play a specified traffic profile.

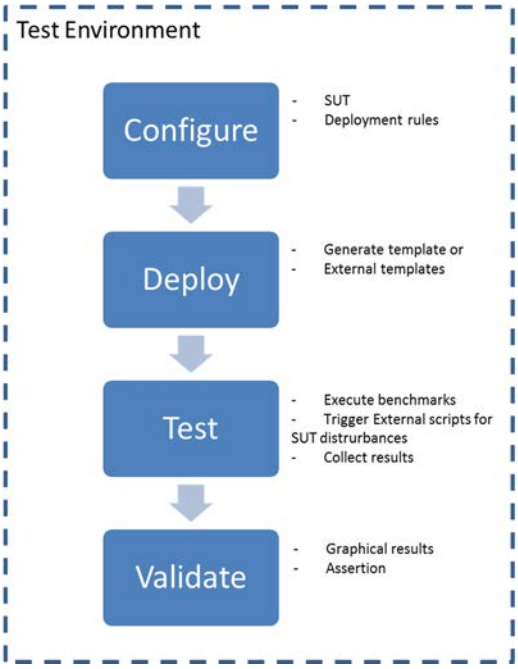

Figure 6.2: Test execution flow for NFVI validation

Test Scenario

A test scenario consists of the particular NFV infrastructure (the SUT) to be validated and the metrics vector representing the VNF type requirements to be validated using the test environment. The metrics vector is defined by the VNF type from which perspective the SUT should be validated.

For the example of a Service VNF type (e.g. CDN controller), the metrics vector will consist of compute, storage and networking metrics relevant to the service executed by the VNF type; for the example of a Network Service VNF type (e.g. Mobile Core control node), the metrics vector will consist of metrics relevant to the service executed by the VNF and also metrics relevant to the routing function. Once the metrics vector is identified, test cases providing measurements are selected. For example, if a critical metric for the service executed by a Service VNF type is IP latency not higher than x milliseconds at all times, a test case implementing a latency measurement application should be executed to verify that the NFV infrastructure is suitable.

Telecom grade VNF applications are expected to provide SLA compliant performance; that depends on consistent behaviour of the NFV infrastructure. Each test case for metric validation should be executed a certain number of times in order to calculate the deviation between the test case results.

The following are examples of test scenarios involving external disturbances injected by the test environment utilizing external scripts:

Latency under infrastructure SW upgrade

- The user defines SLA for latency.
- The test environment configures the system under test (number of virtual machines, amount of memory per virtual machine, amount of virtual cores per virtual machine, network configuration).
- The test environment allows the use of an external script to upgrade the infrastructure SW.
- The test environment starts a test case to measure latency in the configured system.
- The test environment measures latency and verifies SLA fulfilment during infrastructure upgrade.

Overload under high traffic

- The user defines a threshold for frame loss rate.
- The test environment configures the system under test (number of virtual machines, amount of memory per virtual machine, amount of virtual cores per virtual machine, network configuration).

- The test environment allows the use of an external script to generate increasing traffic to the configured system.
- The test environment starts a test case to measure frame loss rate in the configured system.
- The test environment measures frame loss rate and verifies threshold under high traffic.

Packet loss under HW fault

- The user defines a threshold for packet loss.
- The test environment configures the system under test (number of virtual machines, amount of memory per virtual machine, amount of virtual cores per virtual machine, network configuration).
- The test environment allows the use of an external script to disable a compute blade in which a virtual machine is running, simulating a HW fault; the script triggers the creation of a virtual machine to cater for the failed one.
- The test environment starts a test case to measure number of packets lost while the virtual machine is created, active and running.
- The test environment measures the packet loss and verifies threshold under HW fault.

6.4 Reference VNF modelling

The NFV use cases as defined in ETSI GS NFV 001 [i.7] each place specific requirements on, and demands complex configuration from, the underlying infrastructure. In order to verify the infrastructure compliance to those requirements, this methodology decomposes the requirements in metrics, which are represented by test cases.

In order to detect the VNF type and select the applicable metrics and associated test cases for validating an infrastructure in which the VNF application is intended to be deployed, the following are some aspects of the VNF application that needs to be taken into consideration:

- Workload type: User-plane centric or control-plane centric - typically, user-plane centric workload has higher demands on real-time characteristics and delays, whilst control-plane centric workloads are typically compute bound.
- Main components and their requirements on the infrastructure - a VNF application implementation might consist of a number of VNFCs, which could have specific deployment rules, e.g. affinity/anti-affinity; each VNFC realizes a function (or a set of functions) which translates into specific requirements. It is worth noting that the internal realization of the VNFCs is vendor-specific.
- Requirements that implies hardware capabilities - a VNF application implementation might demand capabilities from the infrastructure hardware and related software (e.g. DPDK, SR-IOV, multicore/manycore) which needs to be verified.
- Real time constraints - for user plane-centric, examples of real time constraints are packet drops, response time and synchronization; for control-plane centric, network time-out for cluster communication.
- Hypervisor requirements, as defined in ETSI GS NFV-INF 004 [i.16], such as real-time patches - for user plane-centric, packet processing performance is impacted by interrupt latency variation.
- External interfaces as standardized and their associated requirements which are translated into throughput and latency metrics- for a 3GPP System, the document in ETSI TS 123 002 [i.17] specifies the interfaces, reference points and interfaces towards other networks.

Once the main aspects of the VNF application are identified according to the list above, the next step is to derive the VNF type and the corresponding metrics in order to define the metrics vector.

In order to identify the relevant metrics for a certain VNF type, the below table indicates applicable workload operations for typical VNF types. The list of VNF types covers the most typical Network Functions and each VNF application should be possible to be mapped to one or more of them.

It has to be noted that each individual VNF application does have its own behaviour and the below listed VNF types present common groups and are meant to provide a guideline to identify the relevant metrics. Particular VNF applications might have special requirements which have to be considered in addition to the presented common approach.

The VNF types are specified in accordance to the ETSI NFV use cases ETSI GS NFV 001 [i.7]; the workload operations are specified in accordance to the workloads as in NFV performance and portability best practices ETSI GS NFV-PER 001 [i.9], clause 5.

Table 6.2: VNF type and workload operations

	Data TX/RX	Data session initialization/ termination	Signal processing	Switching/ forwarding	Routing	Data session/flow accounting	Pattern matching	Encrypt / decrypt	H-QoS	Encapsulate/ de-capsulate	Compression	Control session initialization/ termination	Control session management	Access Control/ Authentication	Disk Read/Write
Customer Premises Equipment	✓	✓		✓	✓		✓		✓	✓				✓	
Set-Top Box	✓	✓						✓			✓				
Residential Gateway				✓	✓		✓		✓	✓				✓	
Fixed Access Node			✓	✓	✓				✓	✓					
Radio Access Node	✓	✓	✓	✓	✓			✓	✓	✓		✓	✓	✓	✓
Router				✓	✓		✓	✓		✓				✓	
Session Border Controller							✓		✓			✓	✓	✓	
Firewall				✓			✓			✓					
DPI				✓			✓	✓		✓					
WAN accelerator				✓							✓				✓
VPN				✓				✓							
Mobile Core control node					✓			✓		✓		✓	✓	✓	
Mobile Core user plane node					✓	✓			✓	✓				✓	
Mobile Core User mgmt. node							✓							✓	✓
IMS control node					✓			✓		✓		✓	✓	✓	
IMS user plane node					✓	✓			✓	✓				✓	
IMS Core User mgmt. node							✓							✓	✓
CDN controller					✓								✓		
CDN cache node	✓	✓									✓				✓

Each of the above indicated workload operations do have requirements on certain resources of the NFVI and hence can be validated by metrics from the corresponding sub-groups and categories. Table 6.3 maps the VNF type workload operations to the relevant metrics sub-group and categories.

Table 6.3: VNF workload operations and metric categories

	Compute Performance/ Speed	Compute Capacity/Scale	Compute Reliability/ Availability	Network Performance/ Speed	Network Capacity/Scale	Network Reliability/ Availability	Storage Performance/ Speed	Storage Capacity/ Scale	Storage Reliability/ Availability
Data TX/RX	✓			✓	✓	✓			
Data session initialization/ termination				✓	✓	✓	✓		
Signal processing	✓		✓						
Switching/ forwarding	✓		✓	✓	✓	✓			
Routing	✓	✓		✓	✓	✓			
Data session/flow accounting	✓	✓							
Pattern matching	✓	✓							
Encrypt / decrypt	✓	✓							
H-QoS	✓		✓	✓					
Encapsulate/ de-capsulate	✓	✓							
Compression	✓	✓							
Control session initialization/ termination	✓	✓			✓	✓			
Control session management	✓	✓	✓						
Access Control/ Authentication	✓	✓	✓						
Disk Read/Write		✓		✓			✓	✓	✓

By identifying the applicable workload operations via categorizing the type of the VNF application, it is possible to identify all relevant metric categories. The aggregation of all metrics from the relevant metric categories forms the metric vector for a particular scenario validation.

For example, a VNF application with user-plane centric workload has stringent requirements on delays and packet loss, therefore compute memory latency, storage read/write latency to the disk and networking latency per packet are metrics, as well as networking number of packets sent and received.

From the metrics vector, the applicable test cases are selected to validate the identified metrics. It has to be noted that one test case can be used to validate one to many metrics. Thresholds for metrics, as specified in "Infrastructure Characteristics", here named SLAs, can be specified as reference values for particular metrics to compare against in order to assert if the test case is passed or failed.

For the above stated example, to gather the packet loss, the test case combines two metrics, number of packets sent and received. The test environment triggers the test cases to execute the measurements and assert the SLA values for delay and packet loss.

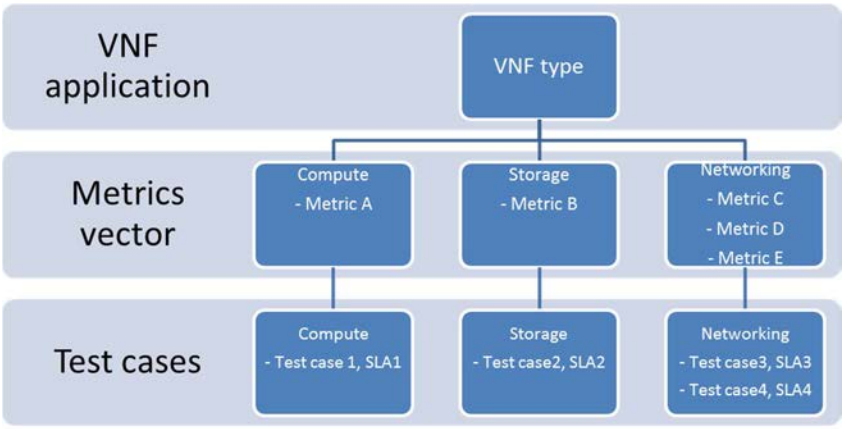

Figure 6.3: Example VNF type, metrics and test case dependency

6.5 Test Case composition

The test environment executes test cases to validate the metrics vector. The test cases representing the individual metrics may be executed individually or combined with other test cases to measure the impact when simultaneously running several test cases. A group of test cases is used to define the typical behaviour of a VNF application. Other combinations of test cases may be used to measure how different VNF types impact each other when running on the same infrastructure.

The combination of simple, generic test cases provides a method for measuring the infrastructure behaviour from the perspective of a VNF application without specifying application implementation details, which are vendor-specific.

The following tables describe generic test cases to be utilized to validate the addressed metrics. The tables contain examples of test tools which could be used to validate the indicated metric but are not meant to be exclusive. Other test tools measuring the same metric could be used to fulfil the same purpose of this methodology.

Additional tests for benchmarking the virtual switches are specified in IETF draft-vsperf-bmwg-vswitch-opnfv-01 [i.10].

The test cases listed below contains examples for some of the metrics stated in clause 6.2 and can be extended by additional test cases on demand.

Network latency test description	
Identifier	NFVI_Validation_L3NetworkLatency_Test_1
Metric	One-way and Round-trip Network Latency
Test Purpose	Measure the L3 network latency between an originating virtual machine and a target entity which can be another virtual machine or a destination outside of the NFVI.
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. This test function acts as origin of the test packets to be sent and contains the latency measurement application as test tool which provides the metric measurements. Dependent on the identified VNF type, a corresponding terminating virtual machine as second test function is deployed on the NFVI according to the deployment constraints of the VNF type. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated. Network latency is measured using round trip delay of sent test requests. SLAs in form of a maximum delay is specified as criterion for this test case.
Test tool example	One-way: OPNFV Project Testing Tools based on IETF RFC 4656 [i.11] Round-trip: TWAMP based on IETF RFC 5357 [i.12]
References	One-way: OPNFV Wiki with Approved Projects: https://wiki.opnfv.org/approved_projects OWAMP [i.13]. Round-trip: TWAMP is based on OWAMP and based on IETF RFC 5357 [i.12].

Network latency test description				
Applicability	Variations of this test case are: • client and server located in different HW stacks (longer communication path, higher latency). • specify a list of different packet sizes. • run test in parallel using different packet sizes. • run test serialized using different intervals.			
Pre-test conditions	The following parameters need to be specified for the latency measurement application: • Packet size. • origin and target IP address. The following parameters need to be specified for the test execution: • Number of test requests. • Interval to send test requests. • Number of times to run the test. • SLAs for maximum One-way and Round-trip delay.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test functions in the NFVI to be validated. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated. The test controller instructs the test functions to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All network latency measurements have been performed.	Latency measurements stored and available for validation
Test Verdict	The network latency measurements are deemed to be acceptable in case they are lower than the required maximum delay SLA.			
NOTE: These examples do not include ping because of the processing time of reflecting host and lack of stream load.				

Network Packet Loss test description	
Identifier	NFVI_Validation_L3NetworkPacketLoss_Test_2
Metric	Number of packets sent, number of packets received, number of packets lost
Test Purpose	Measure the L3 reliability in terms of packet loss between an originating virtual machine and a target entity on the NFVI
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. This test function acts as origin of the test packets to be sent and contains the packet generation tool which provides the metric measurements. Dependent on the identified VNF type, a corresponding terminating virtual machine as second test function is deployed on the NFVI according to the deployment constraints of the VNF type. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated; UDP traffic is configured between the test functions. A SLA in form of a maximum number of packets lost is specified as assertion criterion for this test case.
Test tool example	pktgen
References	IETF RFC 2544 [i.2]; IETF draft-vsperf-bmwg-vswitch-opnfv-01 [i.10]; IETF draft-ietf-bmwg-virtual-net-01 [i.14]; IETF draft-huang-bmwg-virtual-network-performance-01 [i.15]
Applicability	Variations of this test case are: • Test functions on the same server • Test functions on the same NFVI node, different servers • Test functions on the same NFVI PoP, different NFVI nodes • Test functions on different NFVI PoPs • Run the test using different packet sizes

Network Packet Loss test description				
Pre-test conditions	The following parameters need to be specified for the packet generation tool: • Packet size • Distribution of packet sizes • origin and target IP-address • Number of traffic flows (to start the test, e.g. 10) • Packets/sec per flow • Step (in which to increase the number of traffic flows, e.g. 10) The following parameters need to be specified for the test execution: • Duration of the test • SLA for number of packets lost			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test functions in the NFVI to be validated. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated. The test controller instructs the test functions to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All packets sent, packets received measurements have been performed	Number of packets sent, received stored and available for validation.
Test Verdict	The number of packets lost is deemed to be acceptable in case they are lower than the required maximum number of packets lost SLA.			

Network Throughput test description	
Identifier	NFVI_Validation_L3NetworkThroughput_Test_3
Metric	Network throughput
Test Purpose	Measure the L3 network throughput between an originating virtual machine and a target entity on the NFVI.
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. This test function acts as origin of the test packets to be sent and contains the packet generation tool which provides the metric measurements. Dependent on the identified VNF type, a corresponding terminating virtual machine as second test function is deployed on the NFVI according to the deployment constraints of the VNF type. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated; UDP traffic is configured between the test functions. A SLA in form of a target network throughput as monitoring criterion for this test case.
Test tool example	pktgen
References	IETF RFC 2544 [i.2]; IETF draft-vsperf-bmwg-vswitch-opnfv-01 [i.10]; IETF draft-ietf-bmwg-virtual-net-01 [i.14]; IETF draft-huang-bmwg-virtual-network-performance-01 [i.15].
Applicability	Variations of this test case are: • Test functions on the same server. • Test functions on the same NFVI node, different servers. • Test functions on the same NFVI PoP, different NFVI nodes. • Test functions on different NFVI PoPs. • Run the test using different packet sizes.

Network Throughput test description				
Pre-test conditions	The following parameters need to be specified for the packet generation tool: • Packet size. • Distribution of packet sizes. • origin and target IP-address. • Number of traffic flows (to start the test, e.g. 10). • Packets/sec per flow. • Step (in which to increase the number of traffic flows, e.g. 10). The following parameters need to be specified for the test execution: • Duration of the test. • SLA for network throughput.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test functions in the NFVI to be validated. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated. The test controller instructs the test functions to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All packets sent, packets received measurements have been performed.	Throughput in bytes per second stored and available for validation.
Test Verdict	The network throughput is deemed to be acceptable in case is equal to or higher than the required network throughput SLA.			

Storage Performance test description				
Identifier	NFVI_Validation_StoragePerformance_Test_4			
Metric	Input/Output Operations Per Second, storage throughput, storage latency			
Test Purpose	Measure the storage performance in a virtual machine on the NFVI			
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. This test function contains the storage I/O measurement tool which provides the metric measurements. A SLA in form of storage throughput as criterion for this test case.			
Test tool example	Fio (http://freecode.com/projects/fio)			
References				
Applicability	Variations of this test case are: • Run the test for different patterns (e.g. read, read-write, random, sequential).			
Pre-test conditions	The following parameters need to be specified for the storage I/O measurement tool: • File name for workload. • Block size for the IO units. • Type of IO pattern (p.ex. write). The following parameters need to be specified for the test execution: • Duration of the test. • SLA for throughput.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test function in the NFVI to be validated. The test controller instructs the test function to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All input output operations per second, throughput and latency measurements have been performed.	Throughput stored and available for validation.

Storage Performance test description		
Test Verdict	The storage throughput is deemed to be acceptable in case is equal to or higher than the required storage throughput SLA.	

Processor utilization test description				
Identifier	NFVI_Validation_ProcessorUtilization_Test_5			
Metric	Processor utilization			
Test Purpose	Measure the processor utilization in a virtual machine on the NFVI.			
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. The processor measurement tool might be part of Linux Kernel.			
Test tool example	perf stat (https://perf.wiki.kernel.org)			
References	IETF draft-ietf-bmwg-virtual-net-01 [i.14]; IETF draft-huang-bmwg-virtual-network-performance-01 [i.15].			
Applicability	Variations of this test case are: • Use measurement tool for information of context switches, CPU migrations. • Use measurement tool to further characterize the CPU, for example number of CPU cycles, instructions/cycle, Cache misses.			
Pre-test conditions	Configuration of the processor measurement tool, dependent on the tool chosen.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test function in the NFVI to be validated. The test controller instructs the test function to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All input measurements have been performed.	Processor utilization stored
Test Verdict	The processor utilization measurements are stored for reference and comparison. The processor utilization is regarded as secondary metric and not suitable for SLA validation.			

Memory latency test description	
Identifier	NFVI_Validation_MemoryLatency_Test_6
Metric	Latency for random memory access
Test Purpose	Measure the memory latency in a virtual machine on the NFVI.
Configuration	A virtual machine as test function is deployed on the NFVI to be validated. This test function contains the memory benchmarking tool which provides the metric measurements. A SLA in form of memory latency as criterion for this test case.
Test tool example	LMbench
References	IETF draft-ietf-bmwg-virtual-net-01 [i.14]; IETF draft-huang-bmwg-virtual-network-performance-01 [i.15].
Applicability	Variations of this test case are: • Compare performance for different NFVI.
Pre-test conditions	The following parameters need to be specified for the memory benchmarking tool: • Select the latency benchmark (e.g. lat_connect, the time it takes to establish a TCP connection). The following parameters need to be specified for the test execution: • Duration of the test. • SLA for memory latency.

Memory latency test description				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The test controller configures the test function in the NFVI to be validated. The test controller instructs the test function to start the test execution according to parameters defined as pre-test conditions.	
	2	Check	All measurements have been performed.	Memory latency stored and available for validation.
Test Verdict	The memory latency is deemed to be acceptable in case is within the required memory latency SLA.			

6.6 Method for validation

The method for NFV infrastructure validation consists of several consecutive steps. The starting point is the definition of the NFV infrastructure to be validated, followed by the identification of the main aspects of the VNF application from which perspective to validate; this results in the metrics vector. The realization utilizes a test environment for executing the relevant test cases and collecting the results.

Description

Figure 6.4 illustrates the steps involved in the validation execution:

1) Define Infrastructure - represents the SUT, in which a VNF application should be deployed; it includes compute, storage and networking resources.

2) Identify VNF type - relevant aspects of the VNF application targeted for deployment, that should be considered when selecting metrics for the pre-deployment validation of the underlying infrastructure. The VNF type will lead to the selection of the specific metrics (i.e. the metrics vector) to be evaluated.

3) Select Test cases - Based on the selected metrics in step 2), select test cases to validate the metrics.

4) Execute Tests - configure the infrastructure from step 1), deploy the test functions and execute the test cases identified in step 3); test execution includes test stimuli if required.

5) Collect Results - pre-deployment validation of the existing infrastructure for the VNF application defined in step 2).

6) Integrate VNF application - preparation for end to end characterization.

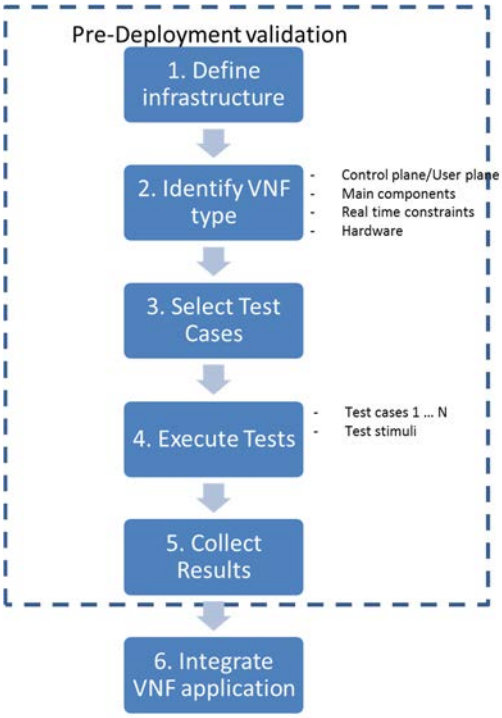

Figure 6.4: NFVI validation execution flow

The implementation of the VNF application, e.g. the number of virtual machines, deployment rules, networking topology, choice of hypervisor, compute, storage and service characteristics is vendor-specific, therefore outside the scope of the NFVI pre-deployment validation methodology. The supporting test environment implements generic test cases for the identified metrics and allows for configuring the system to satisfy implementation specific details.

NOTE: It is to be noted that the execution of the tests in the NFVI pre-deployment validation methodology intends to facilitate the choice of NFV infrastructure for a specific VNF application and to identify possible challenge areas early in the process; it is indeed necessary to subsequently integrate the target VNF application in order to fully characterize the end to end system, as described in step 6. The end to end characterization is outside the scope of the NFVI pre-deployment validation methodology. For the methodology related to characterization of the VNF application, see clause 7.

The purpose of the NFVI pre-deployment validation methodology is to verify the requirements from the VNF applications in the NFV infrastructure; the conformance of the interface implementations themselves and the functional testing of the components are outside the scope of this methodology.

EXAMPLE: The following illustrate the steps of the methodology by means of a concrete example of a VNF application.

Consider the Use Case#5 defined in the clause 9 in ETSI GS NFV 001 [i.7], Virtualisation of Mobile Core Network and IMS. This example will analyse some of the requirements from a user plane node in EPC such as a Packet Gateway (P-GW) on the NFV infrastructure as the criteria for pre-validation. The user plane handling in this example consists of end-user information and associated data transfer control information transported through the user plane.

This example applies the abbreviations and definitions for the basic entities of mobile systems as in ETSI TS 123 002 [i.17] and the Evolved Packet System (EPS) vocabulary as in ETSI TR 121 905 [i.18].

The performance requirements for the EPS are defined in clause 8 in ETSI TS 122 278 [i.19]. The external interfaces specified by 3GPP are defined in clause 6 in ETSI TS 123 002 [i.17].

It is worth noting that references ETSI TS 123 002 [i.17], ETSI TR 121 905 [i.18] and ETSI TS 122 278 [i.19] are used for the purpose of selecting the metrics in the example; compliance and interoperability are not in the scope of this methodology.

Step 1: Define infrastructure.

A concrete NFVI instance comprising of HW, SW and corresponding configuration has been identified as target for the pre-deployment validation. The NFVI instance as SUT is installed, configured and available for all further test execution.

Step 2: Identify VNF type.

The example of a P-GW is identified as a mobile core user plane node VNF type. According to table 6.2, this VNF type includes workload operations such as Routing, Data session/flow accounting, H-QoS, encapsulation/de-capsulation and Access control/authentication.

This goes in line with some of the challenges when defining solutions for this use case as listed in clause 9.5 in ETSI GS NFV 001 [i.7]; among others the fact that services using a network function need not know whether it is a virtual function or a non-virtualised one. To cater to this aspect, the requirements from the external interfaces standardized by ETSI TS 123 002 [i.17] and ETSI TS 122 278 [i.19] on the infrastructure should be target for validation.

According to table 6.3, the identified workload operations map to the metric categories of Performance/Speed, Capacity/Scale and Reliability/Availability of sub-groups Compute and Network. The aggregation of all metrics from those categories according to table 6.1 represents the metrics vector for this scenario.

Step 3: Select test cases.

All metrics from the identified metrics vector can be validated by their corresponding benchmark/test tool. The test cases providing these metric validations represent the set of test cases to be executed for a scenario validation.

Additionally the individual QoS requirements for this VNF application (P-GW) need to be analysed to specify SLA values used for comparison against measured metric values in order to assert the pass or fail of a certain test case.

The following requirement examples, defined by ETSI TS 122 278 [i.19] and the application owner, could be used as reference values that should be fulfilled in order to pass the validation:

- Maximum number of subscribers.
- Uplink packet throughput.
- Downlink packet throughput.
- Latency per flow.

The application specific SLA on maximum number of subscribers could be translated into the metric of maximum number of traffic flows as each served subscriber is represented by its traffic flow.

Step 4: Execute tests.

The test controller of the test environment configures the NFVI under test with additional required configurations. Afterwards the test controller deploys all required test functions containing the benchmarks/tools to execute the test cases for this scenario.

In order to validate the behaviour for this particular VNF type, additional workloads representing this VNF application such as additional P-GWs are deployed on the NFVI by the test controller.

In order to validate the behaviour of the NFVI under load, a traffic generator injecting traffic mixes typical for the S5 interface is started by the test controller.

The test controller of the test environment triggers the execution of the test cases to verify each of the metrics from the metrics vector. If certain test cases require individual external stimuli, these stimuli are also triggered by the test controller.

Step 5: Collect results.

During or after the test case execution (depends on the implementation of the benchmark/tools), the test controller collects the metrics measurement results from the test functions. The metrics measurement results are compared against their SLA values for those metrics where SLA values have been specified. Other metrics measurement results could be aggregated and post-processed to provide benchmarking values.

In case all test cases passed, i.e. all SLA requirements are fulfilled, the NFVI under test is considered to fulfil the requirements of the P-GW VNF application.

7 Pre-deployment validation of VNFs

7.1 VNF lifecycle testing

7.1.1 Introduction

The VNF Manager, in collaboration with the NFV Orchestrator, the VIM and the EM, is responsible for managing a VNF's lifecycle. The lifecycle phases are described in ETSI GS NFV-MAN 001 [i.8] and are listed below:

- VNF on-boarding.
- VNF instantiation.
- VNF scaling/updating.
- VNF termination.

In shared NFV environments, VNFs will likely be instantiated, scaled or terminated at exactly the same instant that many other VNFs are executing in steady state on the same server. This means that one VNF's lifecycle operation can both affect and be affected by the presence or performance of other VNFs executing at the same time, making it essential to thoroughly test the different phases of a VNF's lifecycle. This clause proposes methods and metrics to validate the successful instantiation, scaling and termination of VNFs. Successful VNF on-boarding is considered a pre-requisite and is not addressed in the present document.

7.1.2 VNF instantiation testing

The purpose of this test is to ensure that the VNF Under Test (VNFUT) has been instantiated successfully and is able to perform its network functions upon instantiation. In this test, the VNFUT is bracketed on either end by Test VNFs residing on the same server. The originating Test VNF initiates the appropriate control plane sessions with the newly instantiated VNFUT and exchanges data plane traffic with the VNFUT.

A terminating Test VNF validates that the VNFUT processes and forwards the received traffic correctly. The test methodology and the test topology are shown in figure 7.1.

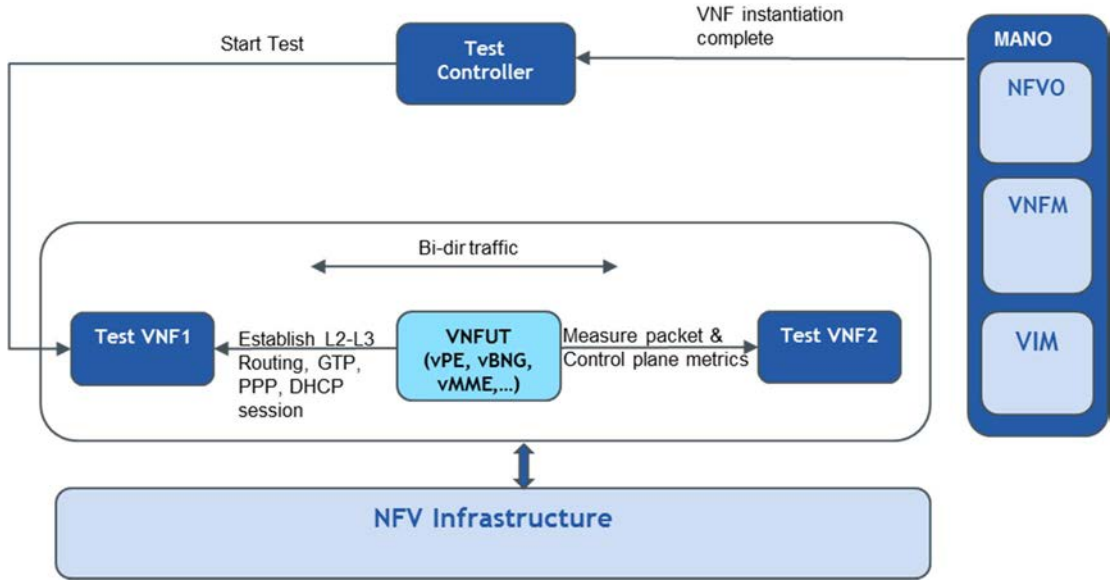

Figure 7.1: Validating the instantiation of VNFs (L2-L3 functions)

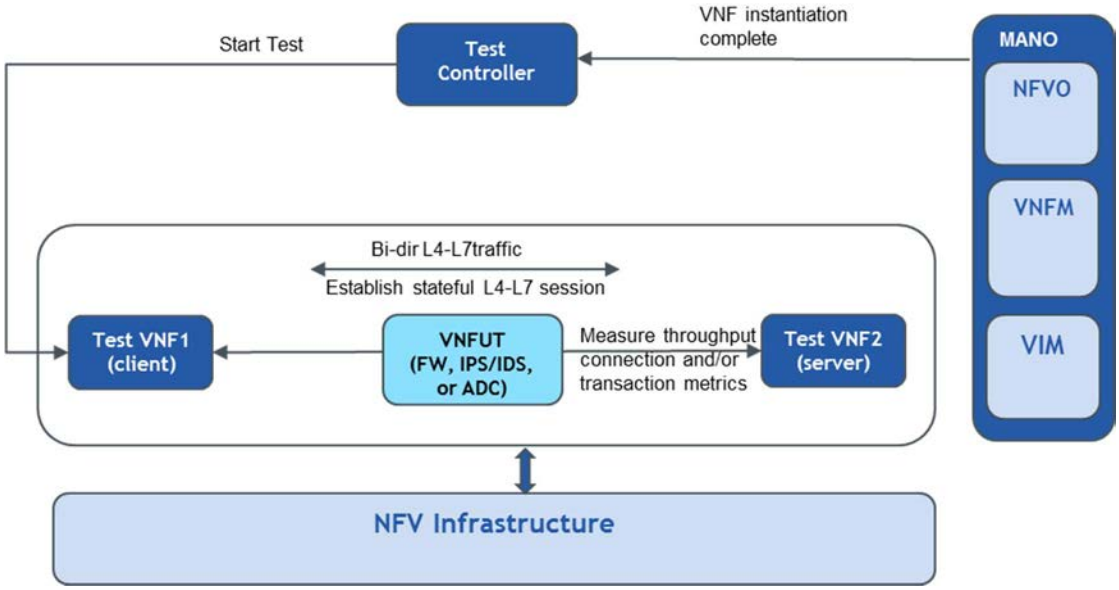

Figure 7.2: Validating the instantiation of VNFs (L4-L7 functions)

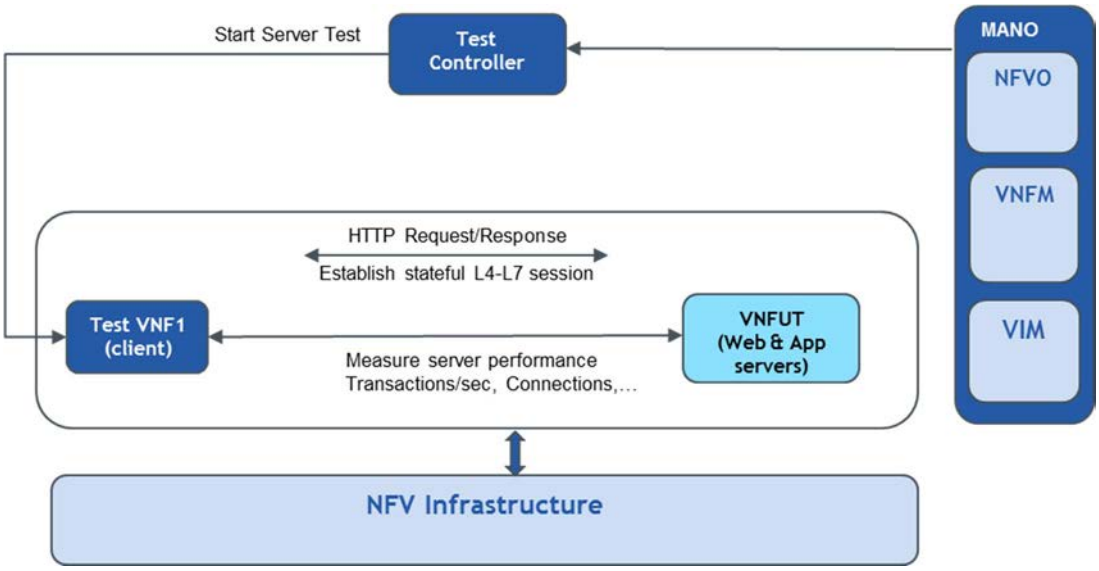

Figure 7.3: Validating the instantiation of VNFs (Application/Web servers)

VNF Instantiation Test Description				
Identifier	VNFB _Instantiation_Test_1			
Test Purpose	To verify that a newly instantiated VNF is 'alive' and functional.			
Configuration	See figures 7.1, 7.2 and 7.3. The flavour of the Test VNF used for the test is dependent on the VNFUT that is being evaluated. For example, an L2-L3 flavour Test VNF is a control plane protocol speaker and is capable of establishing stateful L3 sessions with the VNFUT. A L4-L7 flavour Test VNF can emulate clients and servers and exchange stateful L4-L7 traffic. The VNFUT is surrounded by the Test VNFs and they are connected as a chain for exchanging packets with each other.			
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].			
Applicability	N/A			
Pre-test conditions	• The Test VNFs have been successfully instantiated and configured. • The user has defined the criteria (Ps) for deeming the VNF instantiation as a success. Ps can either be a single metric or a matrix of metrics (for e.g. required forwarding rate, transaction rate, connections per second, etc.). • The user has assigned the necessary NFVI resources for the VNFUT to perform at its target level.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The VNF Manager, in collaboration with the EM and the VIM, triggers the creation and configuration of the VNFUT.	
	2	Check	The VNFUT and its associated VNFCs have been successfully instantiated and have been allocated necessary NFVI resources, as specified in the VNFD The VNFUT is inline between the Test VNFs and is configured with the necessary ACLs, or policies This Check is performed by the EM and/or NFV MANO, and not by the Test VNFs	

VNF Instantiation Test Description				
	3	Stimulus	Test Controller instructs the Test VNFs to initiate the test. Test VNF1 establishes the necessary control plane or stateful sessions with the VNFUT. • If the VNFUT is an L2-L3 device such as a vRouter, vBNG, vMME or vGW, Test VNF1 establishes the necessary control sessions or tunnels with the VNFUT that is needed to exchange traffic user or subscriber traffic. • If the VNFUT is an L4-L7 appliance such as a vFirewall, vIPS, vIDS, vWAN Accelerator or vADC, the Test VNF1 and Test VNF2 establish the necessary stateful TCP/HTTP sessions.	
	4	Check	All the necessary control plane or stateful sessions among Test VNFs and VNFUT have been established.	
	5	Stimulus	The Test VNFs 1 & 2 originate bi-dir traffic toward the VNFUT. • For L2-L3 VNFUT, the originating Test VNFs sends Ethernet, IP or labelled traffic at a user specified rate that will be forwarded by the VNFUT to the terminating Test VNF. • For L4-L7 VNFUT, the Test VNFs originate stateful L4-L7 traffic toward each other that will be forwarded, dropped, or redirected by the VNFUT (based on rules/policies).	
	6	Check	The Test VNFs exchange traffic for at least 10 seconds.	
	7	Check	• The exact liveness checking mechanism and criteria for success are user defined and dependent on the VNFUT. For L2-L3 VNFUT, the Test VNFs ensure that the VNFUT forward all packets without errors, meets its user defined performance targets (≥ Ps) and the Layer 3 state machines are maintained. • For L4-L7 VNFUT, the Test VNFs ensure that the VNFUT correctly processes the traffic (including attack traffic, based on policies), and/or are able to meet its performance targets (≥ Ps) for number of connections, connection setup rate or transaction rate.	
Test Verdict	The VNFUT is deemed as successfully instantiated if all the checks are successful, else it is deemed DoA.			

7.1.3 VNF instantiation in the presence of (noisy) neighbours

The term 'NFV neighbours' refer to other VNFs or services that may compete for the same NFVI resources that are used by the VNFUT. An example of a neighbour is a VNF instance that executes on a separate compute core on the same CPU, where the neighbour and the VNFUT share the same L2 cache.

The purpose of this test is to ensure that the performance and functionality of a newly instantiated VNFUT is not adversely affected by its NFV neighbours. The test also ensures that the newly instantiated VNFUT does not adversely affect the performance of the neighbours already executing. For editorial simplification, the test topology and steps are similar to the test described in clause 7.1.1, with the exception that the NFV infrastructure is shared among an existing VNF as shown in figure 7.4.

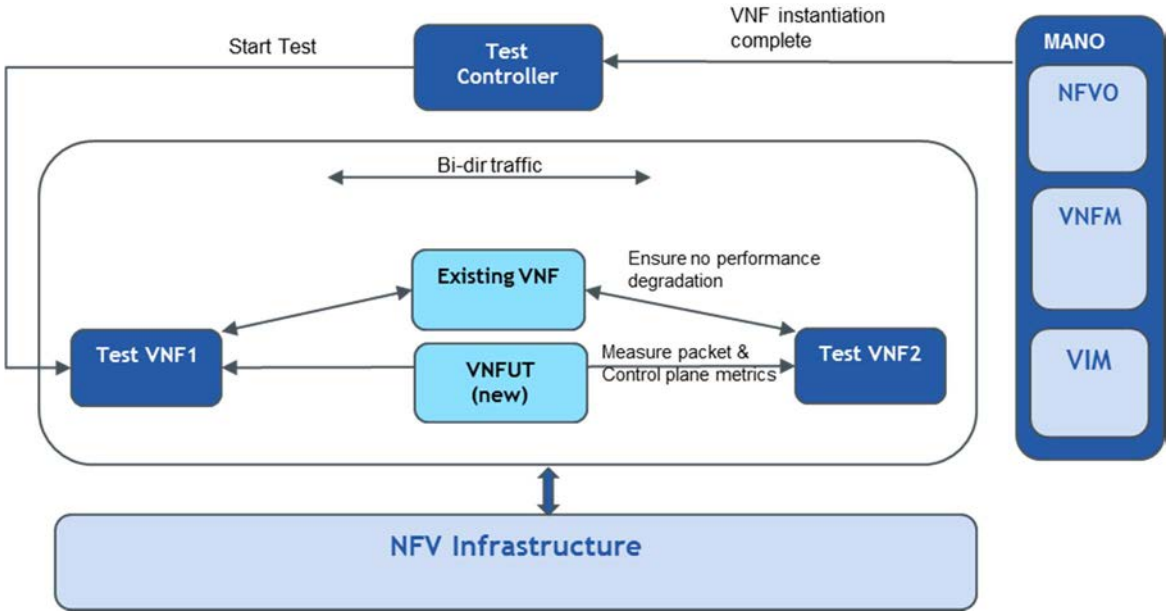

Figure 7.4: Validating the instantiation of VNFs in the presence of neighbours

VNF Instantiation in the presence of neighbours Test Description				
Identifier	VNFB _Instantiation_Test_2			
Test Purpose	To verify that a newly instantiated VNF is not affected by neighbours running on the same shared NFVI.			
Configuration	See figure 7.4. The Test VNFs 1 and 2 are connected to an existing VNF executing on the shared NFVI. The VNFUT is surrounded by the Test VNFs and they are connected as a chain for exchanging packets with each other.			
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].			
Applicability	N/A			
Pre-test conditions	• The user has successfully completed VNFB_Instantiation_Test_1 and validated the successful instantiation of the VNFUT, when tested in isolation. As a result of this test, the user has achieved the target performance level 'Ps' for the VNFUT. • The user has also defined a minimum performance threshold 'Pmin'. When a VNFUT is instantiated in the presence of noisy neighbours, Pmin refers to the minimum acceptable level of performance for the VNFUT. • The existing VNF (neighbour) is connected to test VNFs 1 & 2 and operating at its peak performance level. • VNFUT has been instantiated by the NFVO and configured via its EMS. • The user has assigned the necessary NFVI resources for the VNFUT to perform at its target level. • Test Controller receives trigger from the NFVO that the VNFUT instantiation is complete.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Test Controller instructs the Test VNFs to initiate the test. Test VNF1 establishes the necessary control plane or stateful sessions prior to exchanging traffic with the VNFUT, as described in VNFB_Instantiation_Test_1.	
	2	Check	Validate that the necessary control plane or stateful sessions between Test VNFs and VNFUT have been established.	
	3	Stimulus	The Test VNFs 1 & 2 originate bi-directional traffic toward the VNFUT.	
	4	Check	The Test VNFs exchange bi-directional traffic for at least 10 seconds.	

VNF Instantiation in the presence of neighbours Test Description				
	5	Check	• The Test VNFs ensure that the VNFUT correctly processes and forwards all packets without errors and measures the performance 'P' of the VNFUT. • In addition, the Test VNFs also ensure that the neighbour (existing VNF) continues to perform at its peak performance, during the period when the VNFUT instantiation is being completed.	
Test Verdict	The VNFUT is deemed as successfully instantiated, in the presence of neighbours, if all the checks are successful and the measured performance P ≥ Ps. The VNFUT is deemed as successfully instantiated with a degraded level of performance if the measured performance P is such that Pmin ≤ P ≤ Ps. Else, the VNFUT is deemed as DoA.			

7.1.4 VNF Scaling

7.1.4.1 Introduction

One of the most significant drivers for the transition to NFV is its support for multiple types of VNF scaling. ETSI GS NFV SWA 001 [i.1] has identified three models for VNF scaling, each of which differs based on the functional blocks that are responsible for identifying the trigger for scaling and the issuing of the scaling request.

1) Autoscaling

   - In the autoscaling model, the VNF Manager and theNFV Orchestrator monitor the VNF KPIs and identify the triggers for VNF scaling according to the rules in the VNFD. The scaling request is issued by the NFV Orchestrator. Some examples of VNF KPIs that are monitored are a) NFVI resource utilization, b) user and control plane load, or c) based on events received from the VNF, VIM, EMS or locally generated.

2) On-demand scaling

   - In the on-demand scaling model, the VNF monitors the KPIs of appropriately instrumented VNFCs and triggers scaling by sending a request to the VNF Manager.

3) Manually triggered scaling

   - Scaling is manually triggered by the NOC operators from the OSS/BSS. The manual trigger is typically initiated as a result of an observation of an increased load on the VNF or an expectation of an increased load.

All the three scaling models presented above employ identical scaling mechanisms. Increasing VNF scale is accomplished by scaling out or scaling up. Decreasing or contracting VNF scale is accomplished by scaling in or scaling down.

- When a VNF is scaled out, new VNF components (VNFCs) are instantiated and added to the VNF. Under such circumstances, the VNF may need a mechanism to distribute the load or traffic among the VNFCs (newly instantiated and existing VNFCs). This distribution can be accomplished through the use of load balancers. The load balancer can be a:

  - VNFC that belongs to the VNF that is being scaled; or

  - A separate VNF (e.g. vLoadBalancer) that is instantiated and connects to the multiple VNFCs of the VNFUT.

- When a VNF is scaled in, one or more VNFCs of a VNF are terminated.

- When a VNF is scaled up, it is assigned additional NFVI resources such as compute cores, memory, storage, or network resources.

- When a VNF is scaled down, NFVI resources that have been previously allocated to the VNF are de-allocated.

The test methodology presented in the following clauses only consider the VNF Autoscaling example. The test methods for the validation of the other two VNF scaling models (on-demand and management driven) are identical to the methods described for autoscaling, with the exception of the issuing of the triggers for scaling.

7.1.4.2 Metrics and Methods for validating VNF Autoscaling

This clause presents testing methods and metrics that are needed for validating the completion of VNF autoscaling. It starts with understanding the various flavours of the VNF under test, (as defined in clause 6.1 of ETSI GS NFV-MAN 001 [i.8]) and the conditions that will trigger the dynamic instantiation of a new flavour of the VNFUT. Key metrics such as the time needed to detect Autoscale trigger, time needed to instantiate the new VNF components and distribute the traffic correctly across all the existing and new VNFCs are examined.

1) Understanding the VNF flavours and transitions:

   - The VNFD describes the different VNF flavours, the performance targets for each of the flavours and the Autoscale policies that will trigger the dynamic instantiation of new VNF flavours. It is important to identify the VNF flavour that will be the starting point, its baseline performance and the transitions (to new flavours) that will be tested. Figures 7.5 and 7.6 provide examples of transitions from VNF Flavour A to Flavour B through scale out or scale up. Scale in and scale down will happen in the reverse direction from Flavour B to Flavour A.

Figure 7.5: VNF Autoscaling by instantiating new VNFCs

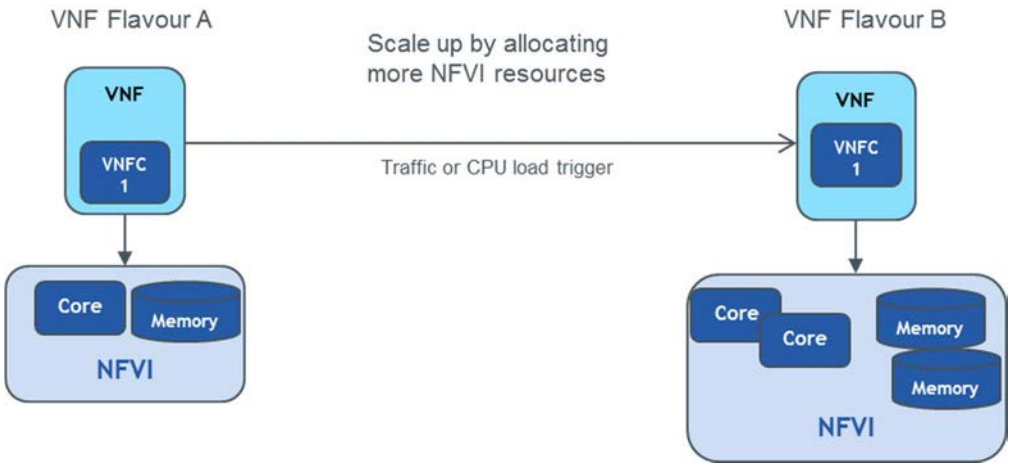

Figure 7.6: VNF Autoscaling by allocating additional NFVI resources

2) Causing the trigger for autoscaling:

   - Autoscaling is triggered based on rules defined in the autoscale policies defined in the VNFD. They can be triggered by VNF specific conditions. Examples of VNF specific conditions are an increase in the number of control plane sessions or user plane traffic above an acceptable limit. They can also be triggered if the utilization of NFVI resources exceeds a threshold for a sustained period of time. Example of NFVI utilization related triggers are CPU utilization, memory utilization, etc.

- Many autoscale triggers have a tight correlation with the traffic load. Examples are increase in a) number of control plane sessions or b) user plane traffic. Other autoscale triggers only have an indirect correlation with the generated load. Examples are NFVI statistics such as a) CPU utilization, or b) memory utilization. The autoscale testing methodology and the timing measurements described in this clause are very accurate only for triggers that are based on traffic load and can be tightly controlled by Test PNFs.

- The timing measurements for autoscaling tests that are triggered by NFVI resource utilization levels are less accurate because the test functions are dependent on NFVI monitoring entities for notifications. For such scenarios, it is recommended that the test be performed multiple times to address the variability concerns associated with the less deterministic autoscale triggers.

3) Time needed to detect the autoscale trigger and complete VNF scaling:

- There are a number of time measurements that are necessary to completely validate VNF autoscaling. Some of these time measurements are externally observable and measurable by test PNFs. There are other time measurements that are not observable by Test PNFs. See figure 7.7.

  - T1, the time when the autoscale trigger is introduced. During pre-deployment validation of VNF autoscaling, it is expected that the test network function will introduce the autoscale trigger.

  - T2, the time when the VNF Manager first detects the Autoscale trigger. This metric is a measure of how quickly the VNF Manager detects the autoscale trigger (after its introduction).

  - T3, the time when the scaling procedures are internally completed by the VNF Manager. This metric is a measure of how quickly new VNFCs or NFVI resources are instantiated/allocated.

  - T4, the time when traffic is first forwarded by the newly instantiated VNFC after a VNF scale out. This metric is a measure of how quickly internal network connections are made between the VNFCs and any internal load balancer.

  - T4[1], the time when traffic is no longer forwarded through a previously instantiated VNFC, after a VNF scale-in.

  - T5, the time when the VNF autoscaling is completed.

- It is recommended that test PNFs be used for validating VNF autoscaling, for two reasons. At the time of writing the present document, microsecond level timing accuracies can only be guaranteed by the use of Test PNFs. Also, the act of making the connections between the VNFUT and Test VNFs inside the NFV server, during autoscaling, can cause distortions to the timing measurements. The distortion can be eliminated by the use of Test PNFs.

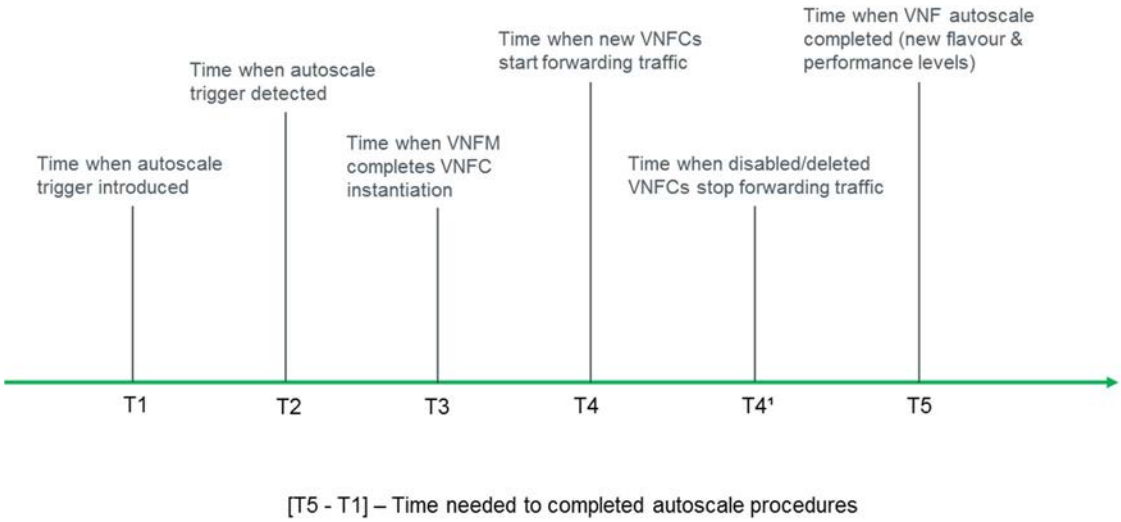

Figure 7.7: Key VNF Autoscaling timing measurements

4) Assessing the VNF performance during Autoscale and post completion of Autoscale:

- It is recommended that the baseline performance of the VNF be established, prior to autoscaling. After the completion of the autoscaling process, the test PNFs benchmark the performance of the new VNF flavour. In addition, it is also recommended to perform a continuous assessment of the VNF during autoscaling. The VNF performance assessment during the transition is essential to ensure that the end users do not suffer severe and unacceptable service degradation.

7.1.4.3 VNF Autoscaling validation

To validate the successful completion of VNF autoscaling, Test PNFs (PNF1, PNF2 and PNF3) are connected to the VNFUT. The Test PNFs initially baseline the performance of the VNFUT (Flavour A, prior to autoscaling), then cause the trigger for autoscaling to Flavour B of the VNFUT and validate the successful completion of autoscale procedures.

The methodology presented below is applicable to a wide range of VNFs and is agnostic to the specific function performed by the VNFUT. Some examples of a VNFUT are L2-L3 VNFs that perform routing, mobility and forwarding functions or L4-L7 VNFs that perform firewall, IDS/IPS or WAN Acceleration functions.

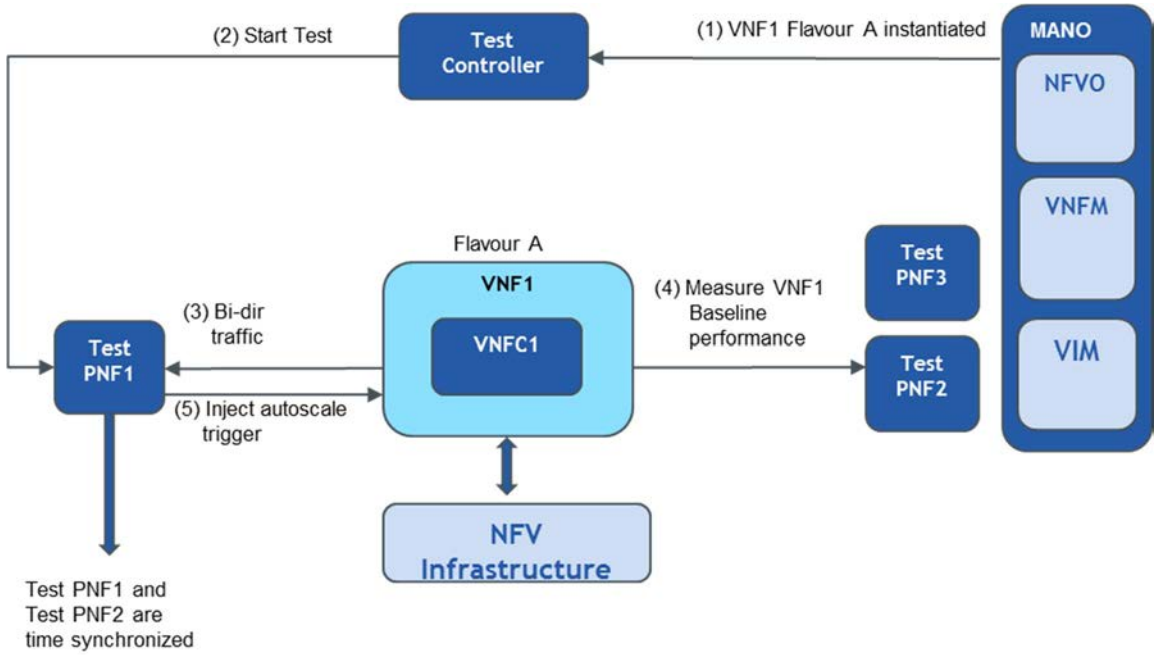

Figure 7.8: Baselining VNFUT prior to Autoscaling (Flavour A)

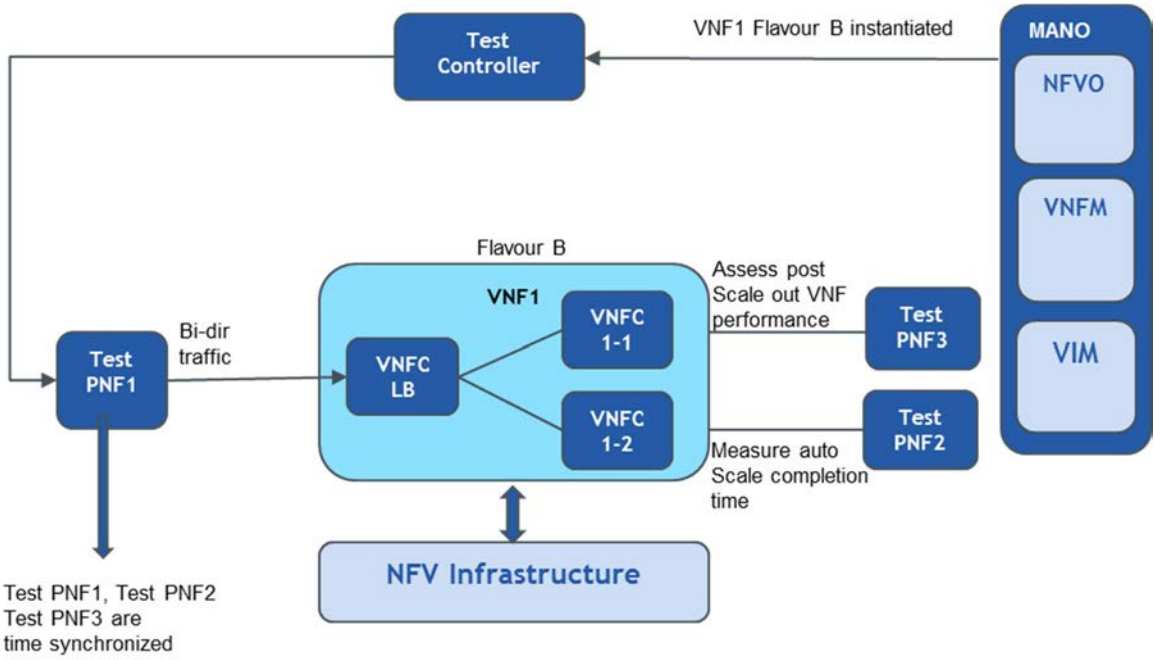

Figure 7.9: VNFUT validation after autoscaling to Flavour B

VNF Autoscaling validation	
Identifier	VNFB_Autoscaling_Test_1
Test Purpose	To verify the successful completion of VNF autoscaling in response to autoscale stimuli. The VNFUT is not the only functional block that is being tested. The MANO components such as the NFV Orchestrator, VNF Manager and the VIM play an active role in the test and are responsible for processing the autoscale stimuli and instantiating VNF components. In effect, the MANO's ability to perform its role in VNF autoscaling is also tested. A non-goal of this test is the validation of the MANO components in isolation or the validation of the interfaces between the VNFUT and the VNF Manager/NFV Orchestrator.
Configuration	See figures 7.7 and 7.8. The Test PNFs 1, 2 and 3 are connected to the VNFUT The VNFUT is connected to the Test PNFs and they exchange bi-directional network traffic with each other.
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].
Applicability	This methodology is applicable to a broad range of VNFs and is agnostic to the specific function of the VNF. Examples include VNFs performing L2-L3 functions such as Routing and VNFs performing L4-L7 functions such as Firewall or IDS/IPS.
Pre-test conditions	• The VNF Provider has defined multiple flavours for the VNFUT. The test starts with Flavour A of the VNFUT. The user has defined the performance target levels for both Flavours A and B of the VNFUT. • The VNFUT has been assigned the necessary NFVI resources to perform at its performance target for Flavour A. • The Test PNFs 1and 2 have the needed resources and capabilities to exchange control and user plane traffic at the performance target levels of the VNFUT Flavour A. • Test Controller is able to access the VNFD and access its fields related to autoscale policy and the stimuli that are needed to cause the autoscaling. • The Test PNFs 1, 2 and 3 have the needed resources and capabilities to stimulate the VNFUT to scale out to Flavour B (increased traffic or CPU load). • The values of Tmax and Tmaxsi, the maximum allowed times for VNF scale out and VNF scale in are known.

VNF Autoscaling validation				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Test Controller instructs the Test VNFs to initiate the test. Test PNFs 1 and 2 establish the necessary control plane or stateful sessions prior to exchanging traffic with the VNFUT Flavour A, as described in VNFB_Instantiation_Test_1.	
	2	Check	Validate that the necessary control plane or stateful sessions between Test VNFs and VNFUT have been established.	
	3	Stimulus	The Test PNFs 1 & 2 originate bi-directional traffic toward the VNFUT Flavour A at its performance target level.	
	4	Check	The Test PNFs 1 and 2 exchange bi-directional traffic for at least 10 seconds.	
	5	Check	The Test PNFs ensure that the VNFUT Flavour A correctly processes and forwards all packets without errors and ensures that the performance 'P' of the VNFUT meets or exceeds its performance target.	
	6	Stimulus	The Test PNFs dial up the traffic and load toward the VNFUT Flavour A, to a level that will trigger a scale out to Flavour B. The exact details of the increase in traffic and load are VNFUT dependent and are outside the scope of this test methodology. The time T1, when the traffic and load reach the level that will trigger autoscaling, is initiated is noted.	
	7	Check	This Check is optional. Note the time T2 when the VNF Manager detects the autoscale trigger and initiates the autoscale from VNFUT Flavour A to Flavour B. The Test PNFs are entities that are external to the VNFUT and the VNF Manager and cannot accurately and independently observe the exact time T2 when the VNF Manager detects the autoscale trigger. Hence, this check is optional.	
	8	Stimulus	From time T1, assess the performance 'P' of the VNFUT periodically by making traffic measurements at Test PNFs 2 and 3. The exact metrics are VNFUT dependent. The polling interval for the VNFUT performance measurements are user defined but it is recommended the polling is done once every second. The polling is done for a user defined maximum of Tmax seconds.	
	9	Check	At the end of every polling interval make the following checks at Test PNFs 2 and 3: • Measure the first instance time t = T4, when traffic is observed on PNF3. Log the value of T4. Do not repeat this step after the T4 has been logged the first time. • Compare the measured performance 'P' of the VNFUT to the performance target for VNFUT Flavour B. If 'P' is lower than the performance target and time t < Tmax, go back to Step 8 and continue polling. • Measure the received traffic at Test PNFs 2 and 3 and ensure that the distribution (load balancing) of traffic across PNF2 and 3 meets the user expectations. If the load balancing is improper and time t < Tmax, go back to Step 8 and continue polling. • If time t > Tmax, go to Test Verdict step. • Else, log time t = T5, as time needed to complete scale out.	

VNF Autoscaling validation				
	10	Stimulus	After time T5 has been logged, reset the timer for the test to 0. From the Test PNFs, reduce the traffic and load in a sustained manner that will cause the VNFUT to scale in from Flavour B back to Flavour A. Log the new time T1, when the traffic and load are reduced to a level that will trigger the VNF scale in. Continue to assess the performance of the VNFUT periodically, at the same polling interval that is used in Step 8.	
	11	Check	This Check is optional. Note the time T2 when the VNF Manager detects the autoscale trigger (for scale in) and initiates the autoscale from VNFUT Flavour B to Flavour A. The Test PNFs are entities that are external to the VNFUT and the VNF Manager and cannot accurately and independently observe the exact time T2 when the VNF Manager detects the autoscale trigger. Hence, this check is optional.	
	12	Check	At the end of every polling interval make the following checks at Test PNFs 2 and 3: • Compare the measured performance 'P' to the performance target for VNFUT Flavour A. If 'P' is higher than the performance target and time t < Tmaxsi, go back to Step 10 and continue polling. • Measure the received traffic at Test PNFs 2 and 3 and ensure that the distribution (load balancing) of traffic across PNF2 and 3 meets the user expectations. Ensure that there are no unexpected disturbances to the received traffic load or unexpected loss of control plane sessions during the execution of the scale in procedure. • In almost all cases it is expected that PNF3 will receive no traffic when scale in is completed. If the load balancing is improper and time t < Tmaxsi, go back to Step 10 and continue polling. • If time t > Tmaxsi, go to Test Verdict step. Else, log time t = T5, as time needed to complete scale in.	
Test Verdict	The scale out of VNFUT from Flavour A to Flavour B is deemed as a failure if the time t > Tmax. The scale out of VNFUT from Flavour A to Flavour B is deemed successful if all checks in Step 9 are successful and the time needed to complete scale out is T5. In addition, present the following metrics to the user: • The value of T1 and T4. • Detailed performance metrics for every polling interval, in the form of a chart during the transition from T1 to T5, highlighting any VNF performance degradation below VNF Flavour A performance target levels. The scale in of VNFUT from Flavour B to Flavour A is deemed as a failure if the time t > Tmaxsi. The scale in of VNFUT from Flavour B to Flavour A is deemed successful if all checks in Step 12 are successful and the time needed to complete scale in is T5. • The value of T1 and T4[1]. • Detailed performance metrics for every polling interval, in the form of a chart during the transition from T1 to T5, highlighting any VNF performance degradation below VNF Flavour A performance target levels.			

7.1.5 VNF Termination

VNF termination is initiated by the NFV MANO. VNFs can be terminated gracefully, during VNF migrations or planned shutdown, or terminated suddenly after encountering an unexpected system or VNF failure. This clause addresses only the graceful termination scenarios and recommends methodologies to validate proper VNF termination. In either case, the NFV Orchestrator is expected to clean up and release any NFVI resources consumed by the VNF and the verification of the proper release of NFVI resources is not addressed.

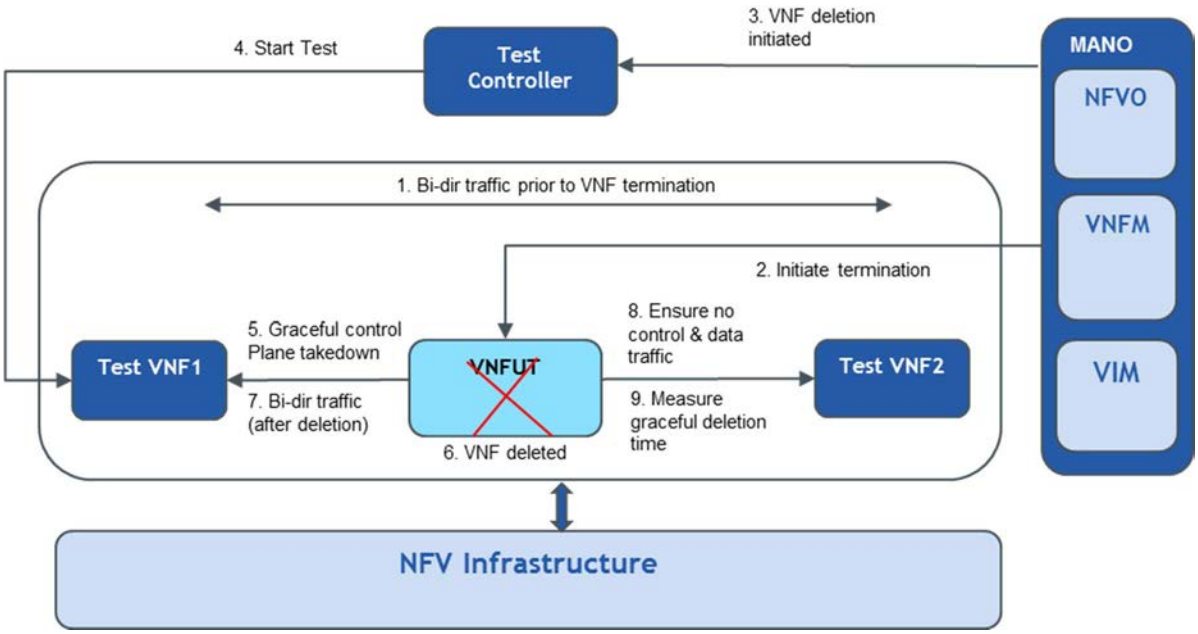

Figure 7.10: Validating VNF termination

VNF Termination Test Description				
Identifier	VNFB _Termination_Test_1			
Test Purpose	To verify that a VNF that is terminated is shut down gracefully.			
Configuration	See figure 7.4. The Test VNFs 1 and 2 are connected to the VNFUT and exchanging bi-directional traffic.			
References				
Applicability	N/A			
Pre-test conditions	• Test VNF1 and Test VNF2 are connected to the VNFUT and exchanging control and data plane traffic.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	VNFM initiates the termination of the VNFUT and informs the Test Controller that the VNF termination has been initiated at time = t1. The VNFUT initiates the graceful takedown of previously established control plane or stateful sessions with Test VNF1.	
	2	Check	Validate that the necessary control plane or stateful sessions between Test VNFs and VNFUT are terminated gracefully and log the time of completion (t2). If the VNFUT has many VNFCs, validate that all the constituent VNFCs and the links between them are terminated gracefully.	
	3	Check	Validate that no traffic is received by Test VNFs 1 and 2 after time = t2.	
Test Verdict	The VNFUT is deemed as successfully terminated if all the checks are successful, else it is deemed as failed.			

7.2 VNF data plane benchmarking

7.2.1 Introduction

The traffic which traverses a VNF is subject to reliability, Quality of Experience (QoE), and predictability requirements. These are defined in the various information elements of the VNFD and stipulated to NFV consumers as a Service Level Agreement (SLA). In order to rigorously evaluate these qualities in a VNF or forwarding graph, data plane benchmarking is required.

Three different types of VNFs are identified. The different VNF types described above have different performance and reliability requirements. The recommended data plane benchmarking methods vary depending on the specific function of the VNF. The frame sizes and frame rates recommended in the test methodologies in clause 7.2 are for illustration purposes. Users are encouraged to pick values that suit their VNF capabilities and deployment needs.

a) VNFs that operate at Layer 2 or Layer 3 and are primarily involved in switching or routing packets at these layers. Examples include vRouter, vBNG, vCE device, or vSwitch.

b) VNFs that operate at Layer 4 through Layer 7 and are involved in forwarding, dropping, filtering or redirecting packets at Layer 4 through 7. Examples include vFirewall, vADC, vIDS/vIPS, or vWAN Accelerator.

c) VNFs that are involved in the dataplane forwarding through the evolved packet core.

7.2.2 Data plane benchmarking of L2-L3 devices

7.2.2.1 Introduction

Data plane benchmarking methodologies for physical L2-L3 devices have been standardized in the IETF and defined in many IETF RFCs:

- IETF RFC 2544 [i.2].
- IETF RFC 2889 [i.3].
- IETF RFC 5180 [i.4].

They are fully applicable and necessary to benchmark virtualised L2-L3 devices in an NFV environment; however, they are not sufficient.

In this clause, additional methodologies necessary to benchmark virtual environments are described and new metrics are defined. They are meant to be used to in conjunction with the IETF RFCs (for example IETF RFC 4271 [i.26] for BGP and IETF RFC 2328 [i.27] for OSPFv2) and not to replace them. Clause 5 describes 2 objectives for a benchmarking test:

i) to validate maximum performance of a VNFUT given a certain level of NFVI resources; and

ii) the amount of NFVI resources needed to attain a certain level of performance for a VNFUT.

For the sake of editorial simplification, the methodologies described in this clause only deal with the first objective, i.e. determining the maximum performance of a VNFUT, for a certain level of resources.

7.2.2.2 Forwarding Performance Benchmarking Test

The topology of this test is similar to the topology discussed in clause 7.1.2. The VNFUT is validated, against basic metrics, under multiple combinations of frame rate and frame size thus forming a grid of results. The presentation of results should take into consideration the underlying hypervisor and the resource efficiency (i.e. cores and memory blocks used by the VNFUT). The intent is to make VNFUT performance comparable across different NFV environments and resource utilization levels.

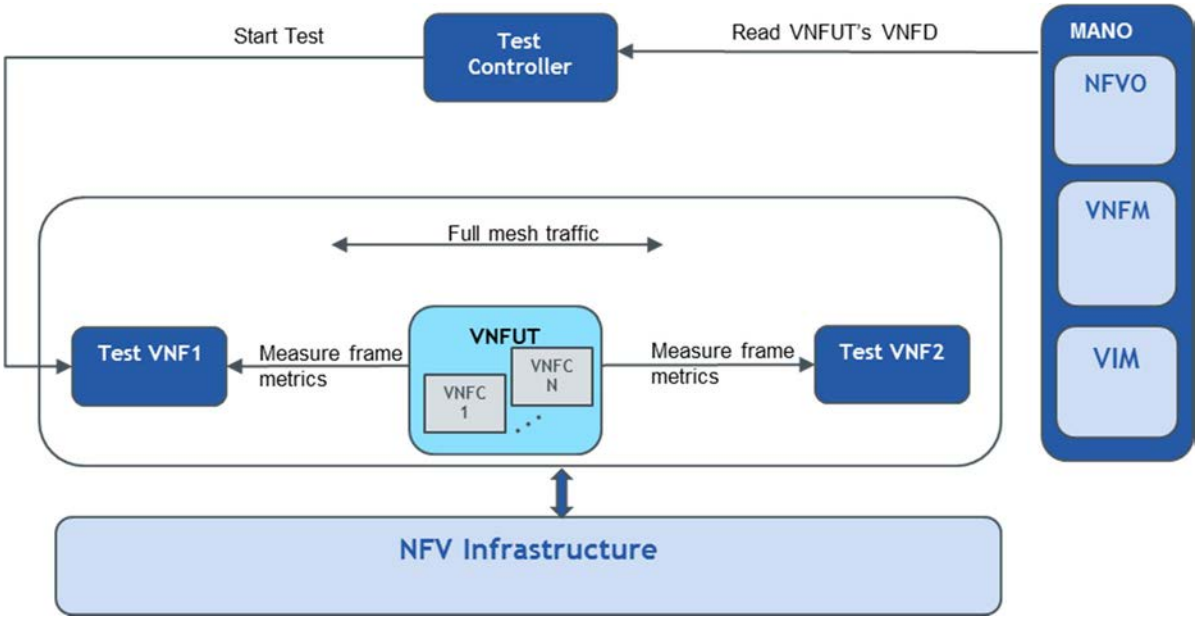

Figure 7.11: Benchmarking the forwarding performance of VNFUT

VNF Forwarding_Performance Test Description				
Identifier	VNFB _L2L3_Forwarding_Perf_Test_1			
Test Purpose	To benchmark a VNFUT's forwarding performance.			
Configuration	See figure 7.11. The Test VNFs 1 and 2 are connected to the VNFUT. The NFVI, VNFM, VIM and NFVO are part of the test environment.			
References	IETF RFC 2544 [i.2], IETF RFC 2889 [i.3], IETF RFC 5180 [i.4] and IETF RFC 5481 [i.20].			
Applicability	L2-L3 VNFs such as vRouter, vCE, vBNG, vSwitch, etc.			
Pre-test conditions	• Test VNF1, test VNF2 and the VNFUT have been successfully instantiated. • The VNFUT is able to receive and process L2-L3 frames from the Test VNFs and forward them.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The Test Controller reads the VNFD of the VNFUT to determine the forwarding performance targets of the VNFUT. The Test Controller instructs the Test VNFs 1 and 2 to start forwarding performance benchmarking tests by originating bi-directional traffic.	
	2	Stimulus	It is recommended that the test cycle through different frame sizes and frame rates. • Frame sizes - [64, 65, 128, 256, 578, 1 024, 1 280, 1 518, 9 022 bytes]. • Frame rates in frames per sec - [10, 100, 1 000, 10 000, 100 000…] The value of the maximum frame rate equals the VNF's forwarding performance target. The exact set of values for frame sizes and rates are dependent on the VNFUT and its performance capabilities.	
	3	Stimulus	Start with a constant frame rate (e.g. 10 fps and cycle through all the different frames sizes. For each frame size, at the current frame rate, send bi-directional traffic from the Test VNF 1 to the Test VNF2 that is forwarded by the VNFUT. It is recommended that each iteration last 120 seconds. Repeat each iteration until all frame sizes and frame rates have been exhausted.	

VNF Forwarding_Performance Test Description				
	4	Check	The following metrics are measured at a user defined polling rate (recommended 1 second) within each measurement pass of an iteration: • Received bandwidth on the Test VNF ports • Sum of sequencing errors (including Frame Loss, Duplicate frames, Out of order frames, Reordered Frames, Late Frames). • Optionally, maximum and average frame delay and frame delay variation are also recorded for each entire pass. See IETF RFC 5481 [i.20] for details. • Further, the utilization of the affiliated cores and memory blocks allocated to the VNF is polled at the same time as measure metrics.	
Test Verdict	For each result within a frame size, the received bandwidth is presented as a set of three values. • The first value of the set is the median bandwidth achieved within the iteration. • The second value presented will be the median bandwidth measured divided by the offered bandwidth expressed as a percentage. • The third metric is the median bandwidth divided by the number of affiliated CPU cores. Maximum latency and maximum packet delay variation also use the same formula for presentation. Absolute metrics like sequence error counts are presented unmodified.			

7.2.2.3 Long duration traffic testing

The purpose of this test is to ensure reliability in shared NFV environments. It is a variant of the "Forwarding Performance Benchmarking Test" in clause 7.2.2.2, and uses the same test setup and analysis. However the tests are run for a longer duration as described below.

VNF Long Duration Test Description				
Identifier	VNFB _Long_Duration_Test_1			
Test Purpose	To verify that a VNF reliably forwards traffic during a long duration test.			
Configuration	See figure 7.9. The Test VNFs 1 and 2 are connected to the VNFUT. The NFVI, VNFM, VIM and NFVO are part of the test environment.			
References	IETF RFC 2544 [i.2], IETF RFC 2889 [i.3], IETF RFC 5180 [i.4] and IETF RFC 5481 [i.20].			
Applicability	L2-L3 VNFs such as vRouter, vCE, vBNG, vSwitch, etc.			
Pre-test conditions	• Test VNF1 and Test VNF2 are connected to the VNFUT. • The "Forwarding Performance Benchmarking Test" described in clause 7.2.2.2 has been completed and a frame size and frame rate is picked for the long duration test.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Pick a specific combination of frame size and frame rate that yielded zero frame loss and run a full mesh traffic test for a long duration. Users are encouraged to pick a duration that matches their deployment needs and VNFUT capabilities but it is recommended that the test runs at least 6 hours.	
	2	Check	Ensure that the VNFUT's performance is consistent throughout the test run. A 1-2 % variation in performance, over time, is acceptable.	
	3	Check	Ensure that the VNFUT has no memory leaks during the long duration test.	
Test Verdict	The VNFUT is deemed as successfully performing on long duration tests if all the checks are successful, else it is deemed as failed.			

7.2.2.4 IMIX Sweep Test

This test is a variant of the "Forwarding Performance Benchmarking Test" test in clause 7.2.2.2, and uses the same test logic and analysis. Instead of fixed frame sizes, IMIX frame size distributions are used. The exact IMIX frame size distributions used in a test depend on the VNFUT and the type of traffic that the VNFUT is expected to encounter post deployment. In the example shown below, three IMIX frame size sets are used with the following mixes of Layer 2 frame sizes:

- IMIX a - [64 bytes (58,33 %), 594 bytes (33,33 %), 1 518 bytes (8,33 %)]
- IMIX b - [90 bytes (58,67 %), 92 bytes (2,00 %), 594 bytes (23,66 %), 1 518 byte (15,67 %)]
- IMIX c - [90 bytes (50,67 %), 594 bytes (23,66 %), 1 518 byte (15,67 %), 9 022 byte (10,00 %)]

To achieve maximal repeatability, each of the sets implement the following frame size sequence, specified according to the IMIX Genome IETF RFC 6985 [i.21], using the custom size designations as follows.

Size, bytes	Custom Code Letter
64	A
90	B
92	C
594	D
1 518	E
9 022	F

IMIX	# Frames	Repeating Sequence
a	70	AAAADD AAADDE AAAADD AAADDE AAAADD AAADDE AAAADD AAADDE AAAADD AAADDE AAAADD AADD
b	50	BBBBBBDDEE BBBBBBDDE BBBBBBDDEE BBBBBBDDE CBBBBBDDE DDE
c	50	BBBDE BBDEF BBBDE BBDEF BBBDE BBDEF BBBDE BBDEF BBBDD BBDDF

Users are encouraged to test with a mix of frame sizes which are matched to their individual deployment, using the methods referenced in IETF RFC 6985 [i.21].

VNF IMIX Traffic Test Description				
Identifier	VNFB_iMix_Traffic_Test_1			
Test Purpose	To benchmark a VNFUT's forwarding performance using an iMix frame distribution.			
Configuration	See figure 7.9. The Test VNFs 1 and 2 are connected to the VNFUT. The NFVI, VNFM, VIM and NFVO are part of the test environment.			
References	IETF RFC 2544 [i.2], IETF RFC 2889 [i.3], IETF RFC 5180 [i.4], IETF RFC 5481 [i.20] and IETF RFC 6985 [i.21].			
Applicability	L2-L3 VNFs such as vRouter, vCE, vBNG, vSwitch, etc.			
Pre-test conditions	• Test VNF1 and Test VNF2 are connected to the VNFUT. • The VNFUT is able to receive and process L2-L3 frames from the Test VNFs and forward them.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The Test Controller reads the VNFD of the VNFUT to determine the forwarding performance targets of the VNFUT. The Test Controller instructs the Test VNFs 1 and 2 to start forwarding performance benchmarking tests by originating bi-directional traffic. Pick an iMix frame distribution that is appropriate to the VNFUT. An example is described in the " IMIX Sweep Test " clause 7.2.2.4.	
	2	Stimulus	It is recommended that the test cycle through different frame rates for the chosen iMix distribution. • Frame rates in frames per sec - [10, 100, 1 000, 10 000, 100 000…]. The value of the maximum frame rate equals the VNF's forwarding performance target.	

VNF IMIX Traffic Test Description				
	3	Stimulus	For each frame rate, send bi-directional traffic from the Test VNF 1 to the Test VNF2 that is forwarded by the VNFUT. It is recommended that each iteration last 120 seconds. Repeat each iteration until all frame rates have been exhausted.	
	4	Check	The following metrics are measured at a user defined polling rate (recommended 1 second) within each measurement pass of an iteration: • Received bandwidth on the Test VNF ports • Sum of sequencing errors (including Frame Loss, Duplicate frames, Out of order frames, Reordered Frames, Late Frames). • Optionally, maximum and average frame delay and frame delay variation are also recorded for each entire pass. See IETF RFC 5481 [i.20] for details. • Further, the utilization of the affiliated cores and memory blocks allocated to the VNF is polled at the same time as measure metrics.	
Test Verdict	For each result, the received bandwidth is presented as a set of three values. • The first value of the set is the median bandwidth achieved within the iteration. • The second value presented will be the median bandwidth measured divided by the offered bandwidth expressed as a percentage. • The third metric is the median bandwidth divided by the number of affiliated CPU cores. Maximum latency and maximum packet delay variation also use the same formula for presentation. Absolute metrics like sequence error counts are presented unmodified.			

7.2.2.5 Flow Misrouting

This test measures the ability of the data plane engine to forward traffic to the right destination. This test uses the same test configuration as shown in the "Forwarding Performance Benchmarking Test" clause 7.2.2.2. The topology is shown in figure 7.12.

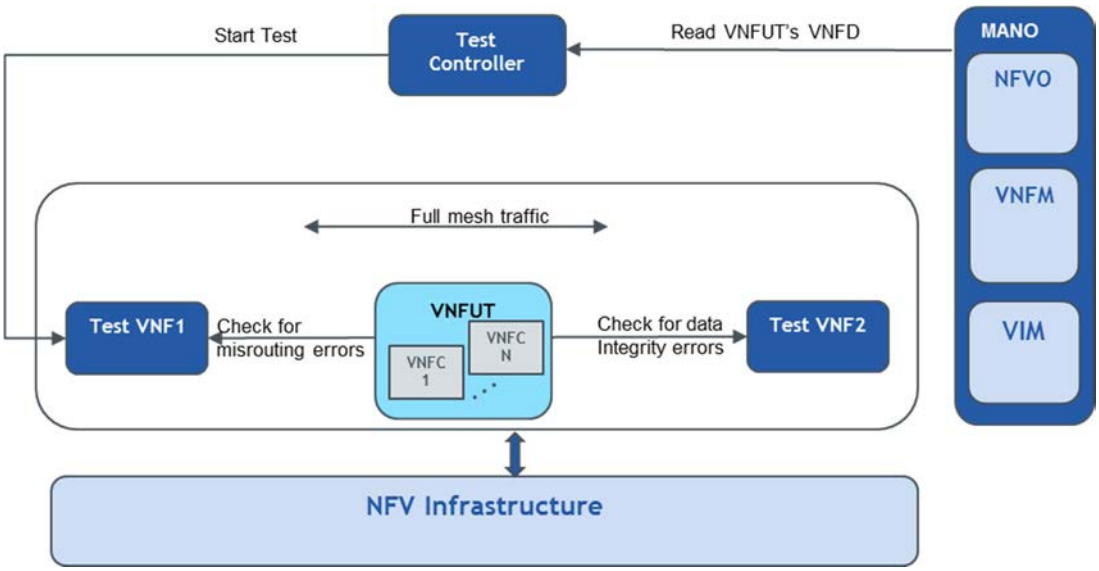

Figure 7.12: Testing for misrouting and data integrity errors in the VNFUT

VNF Flow_Misrouting Test Description				
Identifier	VNFB _Flow_Misrouting_Test_1			
Test Purpose	To verify that a VNF forwards traffic to the right destination.			
Configuration	See figure 7.12. The Test VNFs 1 and 2 are connected to the VNFUT. The NFVI, VNFM, VIM and NFVO are part of the test environment.			
References	IETF RFC 2544 [i.2], IETF RFC 2889 [i.3], IETF RFC 5180 [i.4] and IETF RFC 5481 [i.20].			
Applicability	L2-L3 VNFs such as vRouter, vCE, vBNG, vSswitch, etc.			
Pre-test conditions	• Test VNF1 and Test VNF2 are connected to the VNFUT. • A gratuitous ARP will be required before the first iteration, and those frames will not be included in the results.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Run all the test steps described in 'VNFB _L2L3_Forwarding_Perf_Test_1'. Run all the test steps described in 'VNFB _iMix_Traffic_Test_1'.	
	2	Stimulus	Run a combined test of fixed and iMix frame sizes for each frame rate.	
	3	Check	Ensure that no frame arrives on a vNIC of the Test VNFs that is not expected (a miss switch event).	
Test Verdict	The test is declared as passed if all the checks are successful, else the test is declared as failed if misrouted frames are detected.			

7.2.2.6 Data Integrity Test

This test measures the ability of the data plane engine to forward traffic without data integrity errors. This test uses the same test configuration as shown in clause 7.2.2.5 "Flow Misrouting".

VNF Data_Integrity Test Description				
Identifier	VNFB _Data_Integrity_Test_1			
Test Purpose	To verify that a VNF forwards traffic without data integrity errors.			
Configuration	See figure 7.12. The Test VNFs 1 and 2 are connected to the VNFUT. The NFVI, VNFM, VIM and NFVO are part of the test environment.			
References	IETF RFC 2544 [i.2], IETF RFC 2889 [i.3], IETF RFC 5180 [i.4] and IETF RFC 5481 [i.20].			
Applicability	L2-L3 VNFs such as vRouter, vCE, vBNG, vSswicth, etc.			
Pre-test conditions	• Test VNF1 and Test VNF2 are connected to the VNFUT. • A gratuitous ARP will be required before the first iteration, and those frames will not be included in the results.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Run all the test steps described in 'VNFB _L2L3_Forwarding_Perf_Test_1'. Run all the test steps described in 'VNFB _iMix_Traffic_Test_1'.	
	2	Stimulus	Run a combined test of fixed and iMix frame sizes for each frame rate.	
	3	Check	Ensure that there are no data payload integrity errors. Ensure that there are no L2/L3 errors or CRC errors.	
Test Verdict	The test is declared as passed if all the checks are successful, else the test is declared as failed if data integrity errors are detected.			

7.2.3 Data plane benchmarking of L4-L7 devices

7.2.3.1 Introduction

Benchmarking methodologies for physical L4-L7 appliances such as Firewalls, IPS/IDS, and ADCs have been standardized in the IETF and defined in many IETF RFCs:

- IETF RFC 2647 [i.22].

- IETF RFC 3511 [i.23].
- IETF RFC 6349 [i.24].
- IETF RFC 7230 to IETF RFC 7239 [i.25].

They are fully applicable and necessary to benchmark virtualised L4-L7 appliances in an NFV environment; however, they are not sufficient. In this clause, additional methodologies necessary to benchmark virtual L4-L7 appliances are described and new metrics are defined. They are meant to be used in conjunction with the IETF RFCs (for example IETF RFC 4271 [i.26] for BGP and IETF RFC 2328 [i.27] for OSPFv2) and not to replace them.

7.2.3.2 VNF Application Throughput Test

The goal of this test is to predictably measure the application throughput across a single L4-L7 VNF. Application throughput, also referred to as Goodput (See IETF RFC 2647 [i.22]), is defined as the number of bits per unit of time forwarded to the right destination, minus any bits retransmitted.

As an example, a VNF that processes and forwards application traffic (e.g. vFW, vADC or vIPS/vIDS) is considered for the test. Application throughput will be measured for different HTTP object sizes when using HTTP 1.0, HTTP 1.1 with persistence and HTTP Pipelining.

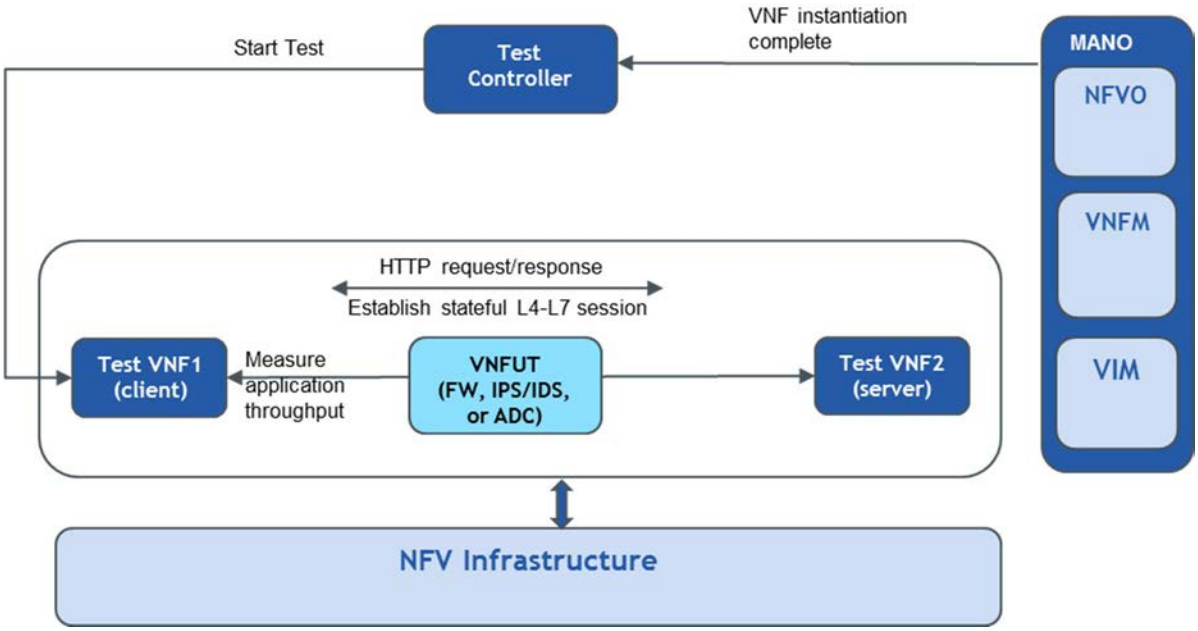

Figure 7.13: VNF Application Throughput Test

VNF Application Throughput Test	
Identifier	VNFB _L4L7_Application_Throughput_Test
Test Purpose	To benchmark an isolated VNF SUT, measuring application throughput using HTTP
Configuration	Refer to figure 7.13 for the test configuration. The Test VNFs 1 and 2 emulate a server pool and client pool. The Test VNFs are connected to the L4-L7 VNFUT.
References	IETF RFC 2647 [i.22], IETF RFC 3511 [i.23] and IETF RFC 6349 [i.24].
Applicability	L4-L7 vFirewall, IPS/IDS, vWAN Accelerator, vADC
Pre-test conditions	• The VNFUT is connected to the Test VNFs 1 and 2 (via vSwitches). • The VNFUT and the Test VNFs have been configured via their EMS. Stateful TCP sessions can be established between the Test VNFs • The VNFUT processes the incoming traffic at Layer 4 through 7 (DPI, NAT, firewall functions, intrusion detection/prevention, etc.).

VNF Application Throughput Test				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	For each HTTP Level {HTTP 1.0, HTTP 1.1, HTTP 1.1 w/ Pipelining} do: For each Object in Set {64 byte, 1K, 10k, 100k, 1M, and 10M} do: • The client (Test VNF 1) issues an HTTP GET for the desired Object. The response is provided by the server (Test VNF 2). • Concurrent requests (by increasing number of TCP connections) are made until the VNFUT cannot forward any more traffic. • The measurement period lasts for at least 300 seconds at the highest number of concurrent requests.	
	2	Check	Per Iteration, the following metrics are recorded: • HTTP Level. • Object Size. • Number of concurrent TCP connects. • The lowest forwarding rate during the 300 second measurement window is recorded as the minimum application throughput across VNFUT.	
Test Verdict	Present the results of all the iterations described above.			

7.3 VNF control plane benchmarking

7.3.1 Introduction

This clause addresses the testing of functions that have only a control plane component, or the control plane component of a function that also accomplishes other types of functionality (like user/data plane).

The metrics to measure for control plane applications are largely different than user plane metrics. Measurements such as packet delay variation and throughput are not relevant in this case. The focus will be mainly on metrics such as scalability of sessions, message processing time and session performance (see ETSI GS NFV 001 [i.7])As always, the discussion of clause 5.3 applies, where configuration of the NFVI fixed and variable parameters will make a marked difference in the metrics being measured.

The VNF control plane benchmarking methodology listed in this clause for MME benchmarking describes a representative use case from a vEPC deployment. The methodology is meant to be generic and to apply to any control plane function.

7.3.2 vMME Control Plane Benchmarking

The purpose of this test case is to benchmark the performance of a single vMME network function. This test determines the VNF's maximum supported session activations/deactivations per second while simultaneously ensuring that the VNF adheres to its maximum allowed utilization of NFVI resources (CPU, memory).

This is achieved by issuing a cycle of Attach Requests, Default Bearer setups and Detaches towards the VNFUT from simulated LTE eNodeBs (Test VNF) at a specific UE event rate. The attach and detach rates are progressively increased through a set of iterations until the maximum performance is obtained. For all iterations, the CPU utilization is periodically measured and ensured that it is within defined limits. The test topology is shown in figure 7.14.

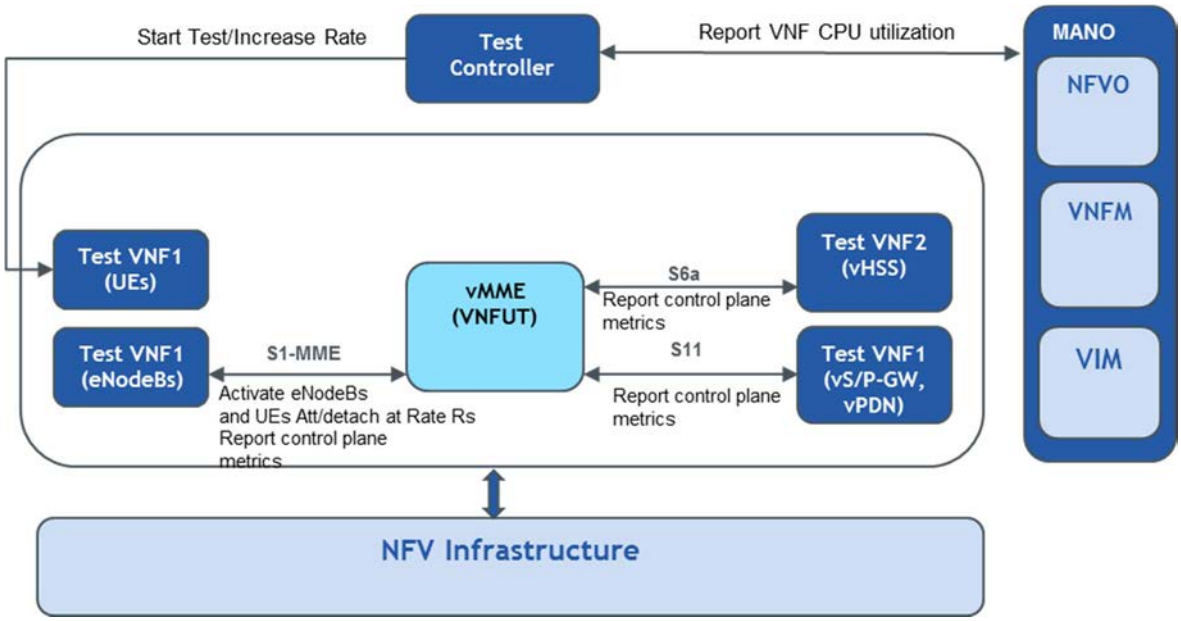

Figure 7.14: vMME VNF Performance Test Topology

vMME VNF Performance Test	
Identifier	VNFB_vMME _Performance_Test_1
Test Purpose	To determine the maximum performance in terms of supported events rate of a single vMME VNF.
Configuration	See figure 7.14. The vMME (VNFUT) will be surrounded by the Test VNFs for full isolation and interface control. Test VNFs can either be installed in the same NFVI and/or can be external to the NFVI. The Test VNF1 will simulate groups of UEs distributed among sets of simulated eNodeBs and will also emulate the SGW to provide a termination endpoint on the S11 interface. The Test VNF2 will emulate an HSS to provide a termination endpoint on the S6a interface.
References	
Applicability	Validation of vEPC components. However, the methodology is equally applicable for benchmarking the control plane performance of any VNF.
Pre-test conditions	• Test VNFs have been dimensioned to support an aggregated rate in excess of the nominal/theoretical rate value of the VFNUT. • Test VNFs have been dimensioned to emulate the necessary number of UEs to support an aggregated rate in in excess of the nominal/theoretical rate value of the VFNUT. • VNFUT has been instantiated by the NFVO and configured via its EMS to accept and handle necessary range of UEs and eNodeBs (e.g. TAC, MNC, MCC, peers list, security mode, S1-release policies, etc.). • The Test VNFs have been successfully instantiated and internally connected to the VNFUT. • 'Mobility event success criteria': The user has defined the maximum performance for a single instance of an vMME in terms of mobile events (for e.g. % of failed attempts, % of dropped sessions, etc.). • 'NFVI success criteria': The user has defined the maximum limits of NFVI utilization for this instance of vMME (for e.g. CPU utilization, CPU max spike). • The user has assigned and isolated the necessary NFVI resources for the VNFUT to perform at its target level. • Test Controller can trigger increase/decrease of UE attach rate as the test progresses.

vMME VNF Performance Test				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Test Controller instructs the Test VNFs to initiate the test and activate their links to the VNFUT: • Test VNF1 emulates eNodeBs, and the S/P-GW. • Test VNF2 emulates the vHSS.	
	2	Check	• All the necessary links and interfaces connecting to the vMME are in active state. • The CPU in 'steady' state is measured and used as the first reference (Cs).	
	3	Stimulus	• Attach Phase: The Test VNFs 1 originates simultaneous UE Attaches towards the vMME at the specified rate R1(attach/sec), until all Attach Attempts have completed. • Hold Phase: Each established session remains active for a brief amount of time. • Detach Phase: Test VNFs 1 originates simultaneous UE Detaches at the specified rate, until all Detach Attempts have completed.	
	4	Check	• Key metrics to analyse: - Mobility: Sessions Attempted, Sessions Established and Sessions Detached. - NFVI: Maximum and Average CPU utilization during Attaches and during Detaches. • Success Case: - Verify that VNF2 vHSS has completed the S6a message flow for as many UEs as defined by 'success criteria' (Sessions Established/Detached). - Verify that VNF1 vSGW has completed the S11 message flow for as many UEs as defined by 'success criteria' (Sessions Established/Detached). - Verify that Test VNF1 UE/eNodeB has completed the S1-MME message flow for as many UEs as defined by 'success criteria' (Sessions Established/Detached). - Verify that maximum CPU utilization during Attach phase has not reached the maximum threshold. - Verify that maximum CPU utilization during Detach phase has not reached the maximum threshold. • Failure Case: - 'Mobility event success criteria' not met at any Test VNF. - 'NFVI success criteria' not met at the VNFUT.	
	5	Check	• If (4) is successful: - Test Controller stores current rate in Rs. - Test Controller stores max measured CPU utilization in Cs. - Test controller indicates Test VNF1 to increase the event rate by 30 % (Rn) and repeat steps (3) and (4). • If (4) is failure: - Test Controller compares current rate with Rs: If Current Rate > Rs, Test Controller indicates Test VNF1 to decrease event rate by 15 % (Rn). Repeat (3) and (4). Else Rs is final.	
Test Verdict	The vMME VNFUT Maximum performance is Rs.			

7.4 VNF control & user plane benchmarking

7.4.1 Introduction

VNFs frequently participate in the processing of both user and control plane traffic. The control plane traffic is associated with the setup of stateful control sessions between nodes and the exchange of session state, routes, subscriber or tunnel information. In most instances, a VNF accomplishes this dual-requirement by instantiating multiple VNFCs, where some VNFCs are responsible for handling control plane traffic and other VNFCs specializing in user plane processing. The VNFCs that handle the control plane processing are expected to be compliant with protocols that are defined in IETF RFCs, ITU, IEEE or 3GPP specifications. Examples include BGP, OSPF, ISIS, LDP, RSVP, etc.

The test method described below ensures that a VNF and its component VNFCs deliver the desired level of both user and control plane performance (by enforcing anti affinity rules) and understanding trade-offs.

7.4.2 vGW's Decoupled Control and User Plane Testing

While the EPC architecture proposes a clear distinction between LTE Control Plane and User Plane, the gateway nodes (vSGW, vPGW or Combo vS/P-GW), should be capable of handling both planes simultaneously in interfaces such as S11 and S5/S8. In today's purpose-built gateways, dedicated resources within the nodes are allocated to these planes in order to guarantee optimal user & control traffic treatment. However, in the context of NFV, the VNFs & VNFCs may have to compete for NFVI resources. A correct implementation of a vGW NF should ensure a continuous and sustained QoS in the user plane regardless of the events occurring in the control plane, and vice versa.

The purpose of this test is to evaluate the performance of the control plane and also the correct allocation of NFVI resources between the control plane and user plane in a vEPC Gateway. In this test, the Test VNFs initiate multiple sessions towards the VNFUT from simulated LTE UE/eNodeBs. While these session are active, the Test VNFs dynamically generate bursts of traffic in the Control Plane; i.e. session activations and deactivations. During these bursts, the User Plane of the vGW will be monitored to ensure the seamlessly handling of these events with no losses in either Control Plane (sessions failed/dropped) or User Plane quality (packets lost/packet delay variation/delayed). This scenario is reflective of the real world traffic encountered in a mobile core network.

The test methodology and the test topology are shown in figure 7.15.

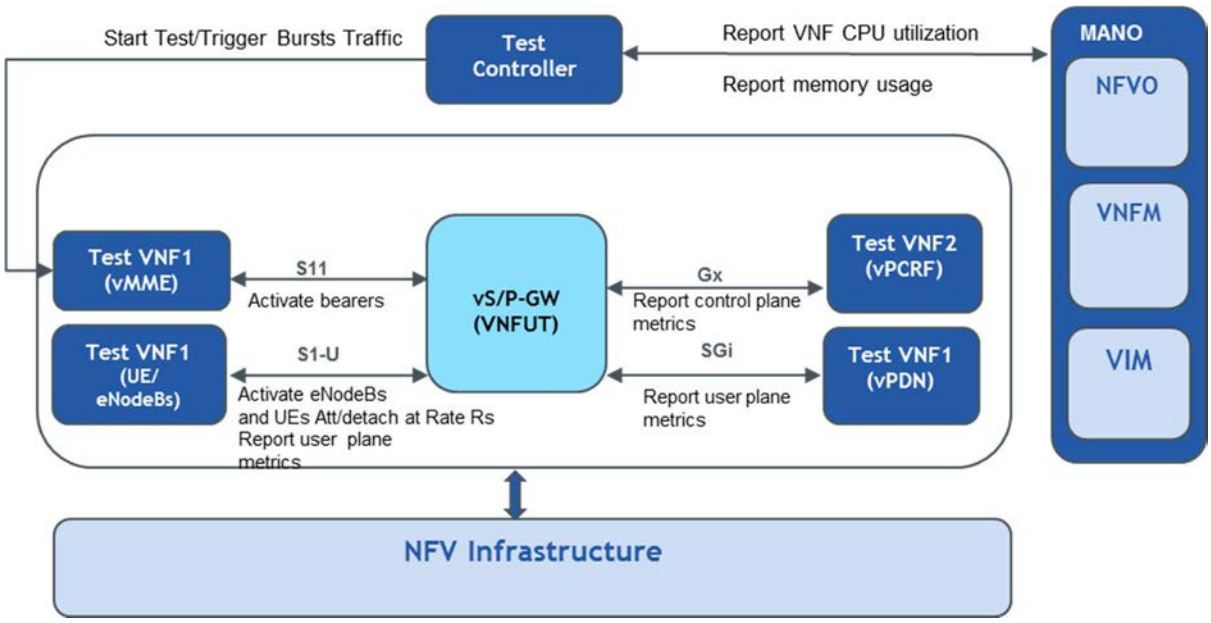

Figure 7.15: vGW Decoupled Control Plane/User Plane Test Topology

v-Gateway Control Plane/User Plane Benchmarking	
Identifier	VNFB_vGW_Control User_Plane_Test_1
Test Purpose	To evaluate correct allocation of NFVI resources between Control Plane and User Plane in a vGW NF for maintaining user plane QoS & control plane scale.
Configuration	See figure 7.15. The vGW (VNFUT) will be surrounded by the Test VNFs for full isolation and interface control. The Test VNF1 will simulate groups of UEs distributed among many eNodeBs and will also emulate the MME and Network Hosts. The Test VNF2 will emulate a PCRF to provide a termination endpoint on the Gx interface.
References	
Applicability	Validation of vEPC components. However, the methodology is equally applicable for benchmarking the control plane performance of any VNF.
Pre-test conditions	• VNFUT has been instantiated by the NFVO and configured via its EMS to accept and handle necessary range of UEs and eNodeBs (e.g, TAC, MNC, MCC, peers list, security mode, S1-release policies, etc.). • The Test VNFs have been successfully instantiated and internally connected to the VNFUT. • Test VNFs 1 has been configured with: - (Static) UE/eNodeB Node/s & S1-U interface simulation with User Plane traffic capabilities. - (Bursty) UE/eNodeB Node/s & S1-U interface simulation with User Plane traffic capabilities. - (Static and Bursty) MME Node & S11 interface simulation with Control Plane call modelling capabilities. - PDN Network Hosts Emulation over SGi. - User Plane line rate traffic generation. • Test VNF1 User Plane traffic per S1-U bearer and SGi interface is stateless 64 bytes UDP packets, in order to stress the VNFUT. • Test VNF2 has been configured as a PCRF Emulator that can support all the necessary UEs, static or bursty, and can trigger bearer activation when necessary. • 'Mobility event success criteria': The user has defined the correct performance for a single instance of an vGW in terms of: - Actual: Activation Rate, Active Subscribers, Active Bearers. - Uplink and Downlink Data Plane Throughput. - Per stream QoS metrics (max and average Latency, Packet Delay Variation, Packet loss). • 'NFVI success criteria': The user has defined the maximum performance for a single instance of an vGW in terms of NFVI resources: - Max number of CPUs. - Max CPU utilization. - Max Memory Storage. • The user has assigned and isolated the necessary NFVI resources for the VNFUT to perform at its target level. • Test Controller has the capabilities to execute the following call model - Static: set up of X Subscribers generating user plane traffic of Y Gbps toward the VNFUT. Hold the sessions opened for the remainder of the test. The value of X and Y depend on the VNFUT and its desired target performance. A suggested value for X and Y are 1M subscribers and 100 gbps, respectively, and are appropriate for today's EPC scale requirements. - Bursty: Trigger a burst of Session Attaches from an additional Z Subscribers. Then trigger a burst of Session Detaches as soon as the UE is connected to the network. Increase the rate of burst and repeat. A suggested value for Z is 800K.

v-Gateway Control Plane/User Plane Benchmarking				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Test Controller instructs the Test VNFs to initiate the test and activate their connections to the VNFUT: • Test VNF1 emulates eNodeBs & vNHs. • Test VNF2 emulates the vPCRF.	
	2	Check	• All the necessary links and interfaces connecting to the vGW are in active state. • The CPU in 'steady' state is measured and used as the first reference (Cs). • The Memory Storage in 'steady' state is measured and used as the first reference (Ms).	
	3	Stimulus	• 'Static' phase: Test Controller triggers the activation of 'X' subscribers at a 'safe' Activation Rate (Rs). • The active UEs proceed exchanging L3-L7 traffic with the emulated Network Hosts. • Sessions are left open for the remainder of the test.	
	4	Check	• All 'static' Sessions are active. • All 'static' Bearers are active and exchanging User Plane traffic. • No Packet Loss (Ps). • Minimized Latency (Ls) < Lth Threshold. • Minimized Packet Delay Variation (Js) < Jth Threshold. • Current CPU Utilization is below threshold and is stored in Cs. • Current Memory Storage is below threshold and is stored in Ms.	
	5	Stimulus	• 'Bursty' Phase: The Test VNF1 vMME Emulator originates simultaneous UE Attaches towards the vGW at the specified rate Rb(attch/sec), until all Attach Attempts have completed. Each established session remains active for a brief amount of time. Then Test VNF1 vMME originates simultaneous UE Detaches at the specified rate, until all Detach Attempts have completed.	

v-Gateway Control Plane/User Plane Benchmarking				
	6	Check	• Key metrics to analyse: - Mobility: Bursty Sessions Attempted, Bursty Sessions Active, Bursty Sessions Detached, Static Sessions Active, Packet Loss, Latency, Packet Delay Variation. - NFVI: Maximum CPU utilization during bursty traffic. Maximum Memory Storage during bursty traffic. • Success Case: - Test VNF2 vPCRF has completed the Gx message flow for as many UEs as defined by 'success criteria' for bursty sessions (Sessions Established/Detached). - Test VNF1 vMME has completed the S11 message flow for as many UEs as defined by 'success criteria' for bursty sessions (Sessions Established/Detached). - Test VNF1 UE/eNodeB has completed the S1-U message flow for as many UEs as defined by 'success criteria' for bursty sessions (Sessions Established/Detached) - Test VNF1 NH reports no packet loss (Ps), latency below threshold (Ls) and Packet Delay Variation below threshold (Js). - Test VNF1 reports no 'static' sessions dropped. - Maximum CPU utilization during 'bursty' phase has not reached the maximum threshold. - Maximum Memory Storage during 'bursty' phase has not reached the maximum threshold. • Failure Case: - 'Mobility event success criteria' not met at any Test VNF. - 'NFVI success criteria' not met at the VNFUT.	
	7	Stimulus	• If (6) is successful: - Test Controller stores current 'bursty' rate in Rb. - Test Controller stores max measured CPU utilization in Cs. - Test Controller stores max measured Memory Storage in Ms. - Test Controller indicates Test VNF1 vMME to increase the event rate by 30 % (Rn) and to repeat steps (5). • If (6) is a failure: - Test Controller compares current rate with Rb: o Current Rate > Rb, Test Controller indicates Test VNF1 vMME to decrease event rate by 15 % (Rn). Repeat (5). o Else Rb, Cs, Ms are final.	
Test Verdict	The vGW Max CPU Utilization, Memory Storage and Units for User Plane QoS sustainability are Cs,Ms and a bursty traffic at Rb.			

8 Pre-deployment validation of Network Services

8.1 Introduction

A network service (NS) comprises a chain of service functions (forwarding graph of virtual or physical NFs).

The NFVO, in collaboration with the VNF manager, the NFVI, the VIM, and the OSS/BSS, manages the lifecycle of one or more Network Services. The NFVO has the end-to-end view of resource allocation and serves as the single point of access for all requests from the OSS. In a shared NFV environment, multiple network services (implemented as VNF service chains) are executing on the same physical infrastructure, each at its own stage of the lifecycle. Some will be instantiating, scaling, or terminating while others are executing in a steady state. Lifecycle testing is essential to determine whether lifecycle changes of one NS is affecting other NS.

The test methodologies proposed in this clause describe methods to validate the successful instantiation of NS, the speed of instantiation of NS and the scaling of NS. It is assumed that the constituent VNFs of the Network Service have already been validated prior to the execution of the Network Services testing. It is NOT the goal of this clause to describe protocol specific testing methods for validating VNFs such as a vFirewall, vADC, vWOC, vCPE, vBNG or vPE.

Network Services can be deployed in an NFV environment in one of the following 3 methods:

- The VNF forwarding graph completely originates and terminates within the same NFV server.
- The VNF forwarding graph originates in an NFV Server A and terminates in an NFV server B, where both the servers are within the same data centre.
- The VNF forwarding graph originates in NFV server A, is connected to physical network functions across a WAN (PNFs), and terminates on an NFV server B in another data centre.

The methodologies covered in the following clause specifically address use cases where the VNF forwarding graph originate and terminate within the same server; however, these methodologies are generic and are equally applicable to the other two deployment methods described above.

8.2 Network Services -Instantiation testing

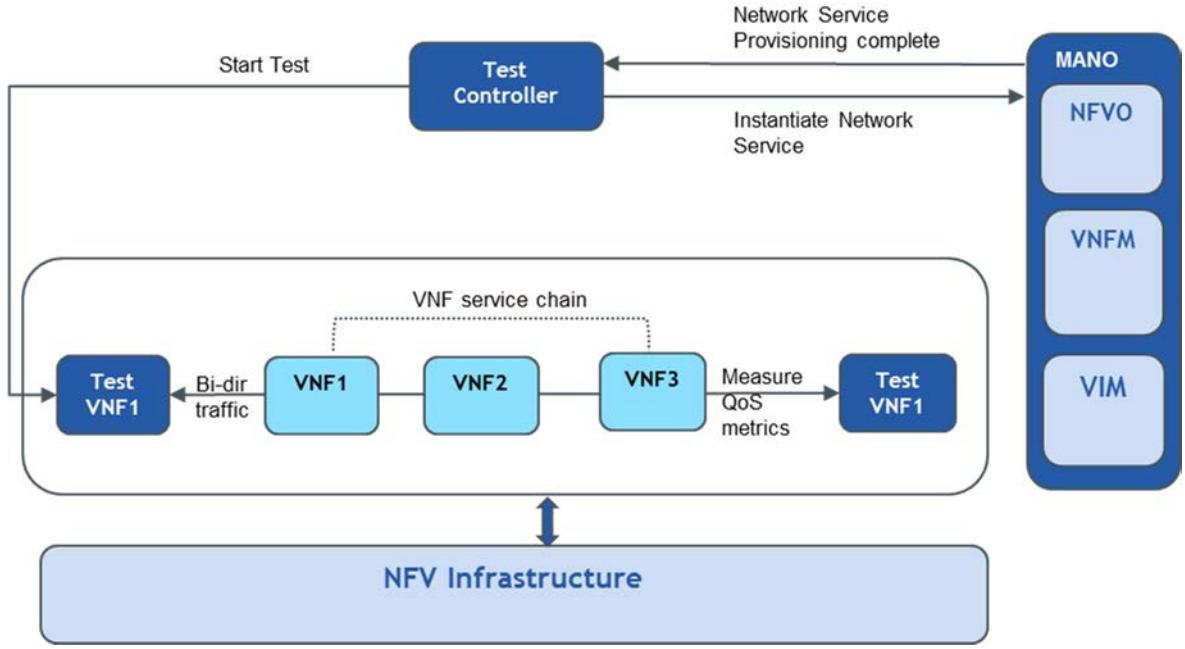

Figure 8.1: Network Services - Instantiation testing

Network Services Instantiation Test				
Identifier	NS_Instantiation_Test_1			
Test Purpose	To verify that a newly instantiated Network Service is 'alive' and functional.			
Configuration	See figure 8.1. The test setup consists of an NFV server hosting a Network Service under Test (NSUT). An example of such a Network service would be a vCPE service chain consisting of a vFirewall, vCE device and a vRouter. Virtual test devices are connected to the NSUT and will be used to originate and terminate traffic. The flavour of the Test VNF used for the test is dependent on the NSUT that is being evaluated. In this example, an L4-L7 flavour Test VNF is used to validate the vCPE service chain.			
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].			
Applicability	vCPE, vEPC (VoLTE & vIMS) and many other NFV use cases.			
Pre-test conditions	• The VNFs of the NSUT have been dimensioned in a manner that the test VNFs do not compete for the same NFVI resources as the VNFs of the NSUT. If such a configuration cannot be accurately enforced, physical test devices should be used, to prevent the test device from skewing the performance of the NS. • The user has defined the criteria (Ps) for deeming the NS instantiation as a success. Ps can either be a single metric or a matrix of metrics (for e.g. required goodput rate, connections per second etc.). The exact set of criteria is dependent on the Network Service under consideration. • Perform validation of the NFVI as specified in clause 6. • Perform validation of each constituent VNF of the NS as specified in "VNF instantiation testing".			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The Test Controller instructs the NFV Orchestrator to instantiate the Network Service. In some cases, the VNFs that constitute the NS may already be instantiated and instantiating a NS may only involve interconnecting the VNFs as defined by the NSD. In other cases, the instantiation of a NS may involve the additional steps of instantiating the constituent VNFs prior to interconnecting them. Please refer to clause C.3 of ETSI GS NFV-MAN 001 [i.8] for detailed NS Instantiation flows.	
	2	Check	Verify that the NFV Orchestrator notifies the test Controller after it completes the NS instantiation. If the Orchestrator indicates success, proceed to Step 3, Else, skip to Test Verdict and indicate DoA.	
	3	Stimulus	The test VNFs exchange appropriate bi-directional L2-L7 traffic with the Network Service under Test. • For example, a vCPE service function chain consisting of a vFirewall, vCE device and a vRouter will receive a realistic traffic mix of HTTP (large and small objects), FTP and DNS traffic. • Bi-directional traffic is originated from the Test VNFs at a rate that matches the performance target (Ps) of the NSUT.	
	4	Check	• Ensure that the Test VNFs exchange service frames for at least 10 seconds. • Measure the Service performance (QoS) metrics and ensure that the QoS metrics meet or exceed the target performance (Ps) of the Network Service.	

Network Services Instantiation Test				
	5	Stimulus	Repeat steps 1 through 4 for the different flavours of the Network Service and variations of the traffic profile: • Use traffic profiles that include IPv4 and IPv6 traffic and a mix of IPv4 and IPv6. • Variations in the service flavour and traffic profile. For example, a) use of a minimal set of firewall rules or b) a large set of firewall rules that also include intrusion detection and DPI.	
Test Verdict	The Network Service is deemed as successfully instantiated if all the checks are successful. Else, the Networks Service is deemed as a Degraded Instantiation with sub-SLA performance, and reasons should be investigated and reported. Else, the Network Service is deemed as DoA.			

8.3 Network Services - Speed of activation

One of the biggest promises of NFV is the ability it provides operators to activate services, when needed, and terminate network services, when not needed. These promises are enabled by the ability of the NFV Orchestrator to quickly on-board and activate services, that may be implemented as a chain of VNFs. The speed of activation of the Network Services has a major influence on the end user QoS. It is dependent on the near instantaneous communication between the various NFV components such as VIM, NFVO and VNFM and ability to quickly setup connections between the constituent VNFs. It is also influenced by the number of Network Service instances that are already provisioned and executing on the NFV environment. An NFV server operating at low utilization is likely to turn up services faster than NFV servers operating at capacity.

To ensure end-user QoS, it is necessary to validate the speed of activation of services in highly dynamic NFV environments. For accurate measurement of time, the use of physical test devices (that are synchronized with each other at microsecond accuracy) are recommended. Using physical test devices also eliminate any effect that the test functions may have on the test measurements. The test topology is shown in figure 8.2.

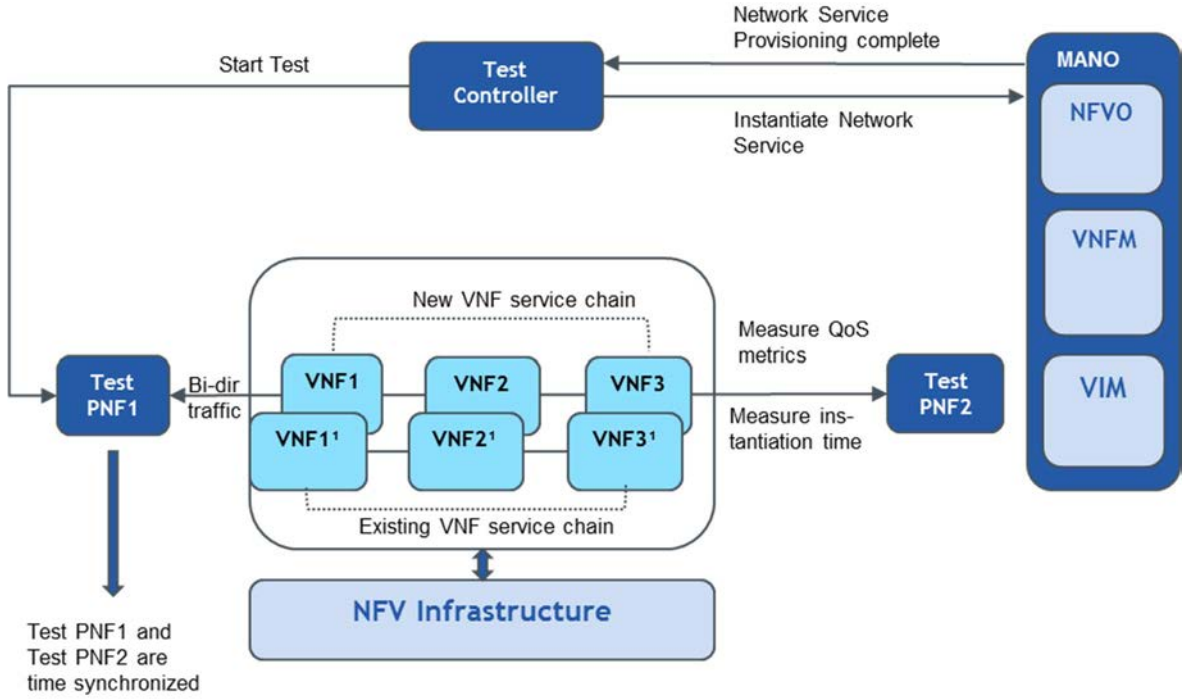

Figure 8.2: Network Services - Measuring speed of activation

Network Services Speed of Activation Test				
Identifier	NS_Speed_of_Activation_Test_1			
Test Purpose	To measure the time needed to activate a Network Service that comprises multiple VNFs in a service chain.			
Configuration	See figure 8.2. The test setup consists of an NFV server hosting a Network Service under Test (NSUT). The NSUT is comprised of 3 VNFs in a service chain. An example of such a Network service would be a vCPE service chain consisting of a vFirewall, vCE device and a vRouter. Physical test devices are connected to the NSUT and will be used to originate and terminate traffic.			
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].			
Applicability	vCPE use case. vEPC use cases (VOLTE and IMS services).			
Pre-test conditions	• The user has defined the criteria (Ps) for deeming the NS instantiation as a success. Ps can either be a single metric or a matrix of metrics (for e.g. required goodput rate, connections per second etc.). The exact set of criteria is dependent on the Network Service under consideration. • The user has defined Tmax as the maximum time allowed for the completion of Network Service activation. • Perform validation of each constituent VNF of the NS as specified in clause "VNF instantiation testing". • Perform validation of NS Instantiation as specified in clause "Network Services - Instantiation testing". • Physical test devices are used, to ensure microsecond accuracy in timing measurements and to eliminate any influence that the presence of a test VNF will have on the shared NFV environment			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The Test Controller instructs the NFV Orchestrator to instantiate the Network Service at time T = T1. In some cases, the VNFs that constitute the NS may already be instantiated and instantiating a NS may only involve interconnecting the VNFs as defined by the NSD. In other cases, the instantiation of a NS may involve the additional steps of instantiating the constituent VNFs prior to interconnecting them. Please refer to clause C.3 of ETSI GS NFV-MAN 001 [i.8] for detailed NS Instantiation flows.	
	2	Check	Verify that the NFV Orchestrator notifies the test Controller after it completes the NS instantiation. If the Orchestrator indicates success, proceed to Step 3, Else, skip to Test Verdict and indicate DoA.	
	3	Stimulus	At time T = T1, at the same time that the Network Service instantiation is requested, the Test Controller instructs the test devices to start exchanging appropriate bi-directional L2-L7 traffic with the Network Service under Test. • Traffic is originated at a rate that matches the performance target (Ps) of the NSUT.	
	4	Check	• Measure the service performance (QoS) metrics periodically (recommended once every 100 ms) until the time when all the QoS metrics meet or exceed the target performance (Ps) of the Network Service. Log the time T = T2 when service performance ≥ Ps. • If T2-T1 > Tmax, skip to Test Verdict step and indicate Network Service Activation failure. • Else, log [T2-T1] as the time needed to complete the Network service activation.	

Network Services Speed of Activation Test				
	5	Stimulus	Repeat steps 1 through 4 for the following variations: • Use multiple frame sizes and traffic profiles that include IPv4 and IPv6 traffic and a mix of IPv4 and IPv6. • Vary the 'loading' of the NFV server. Start the first pass by activating the service chain when there are no other service chains provisioned in the NFV server. For additional test passes, activate new service chains, when there are service chains that are already executing in the same NFV server. • Repeat test steps 1 through 4 for each service deployment flavour as defined in the NSD. For example, if there are three flavours defined, 3 different NS instances will be instantiated and each of them will be subjected to speed of activation test steps as described above.	
Test Verdict	If the checks in step 4 are successful, plot [T2-T1] as the time needed for Network Service activation for each variation of the test, as described in step 5. Else, Network Service activation is deemed as failed for this test variation, and the reasons for the failure should be investigated.			

8.4 Network Services - Autoscaling validation

One of the most significant drivers for NFV is the ability it provides to network operators to allocate resources when needed to meet dynamically increasing network demands and contract resources when not needed.

Clause 7.1.4 "VNF Scaling" describes three models for VNF scaling, each of which differ on the method for triggering the scaling action. These three models, a) Autoscaling, b) On-demand scaling and c) Manually triggered scaling are equally applicable to Network Services scaling too.

A Network Service can dynamically react to a sustained spike in customer traffic by scaling out; similarly, during periods of reduced customer traffic, it can scale in. The scaling out and scaling in of Network Services are briefly described below:

- Network Service scale out - NS Scale out is accomplished by one of the following three methods:
  - By instantiating new constituent VNFs;
  - By instantiating new VNF components (scale out); or
  - By increasing NFVI resources assigned to VNFs (scaling up).
- Network Service scale in - NS Scale in is accomplished by one of the following three methods:
  - By terminating existing constituent VNFs;
  - By terminating existing VNF components (VNF scale-in);
  - By de-allocating NFVI resources that were previously assigned to existing constituent VNFs (VNF scale sown).

Figures 8.3, 8.4 and 8.5 conceptually depict the scale out of a Network Service from Flavour A to Flavour B.

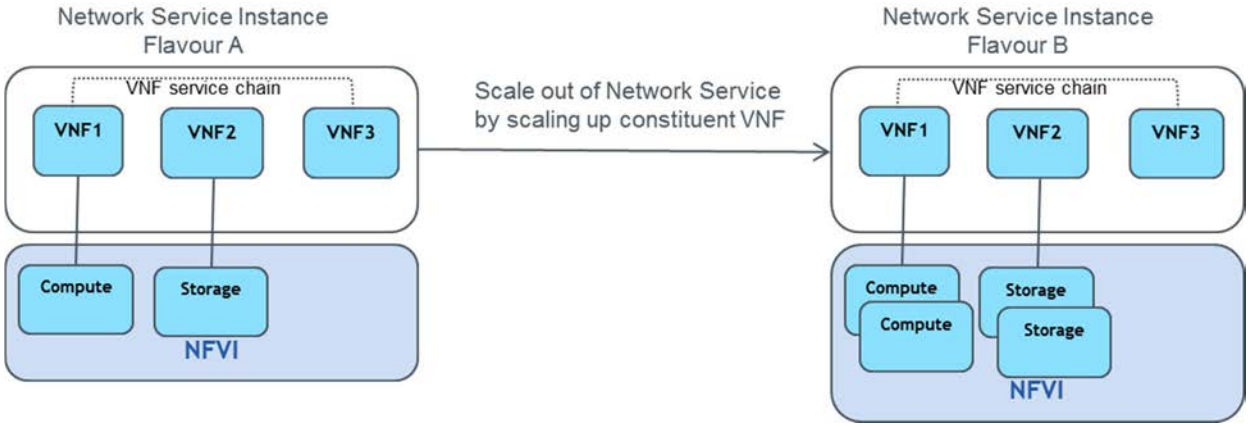

Figure 8.3: Network Services Scaling - Scale out by allocating more NFVI resources

Figure 8.4: Network Services Scaling - Scale out by allocating new VNFs

Figure 8.5: Network Services Scaling - Scale out by allocating new VNF components

The methodology presented in this clause addresses autoscaling only and highlights a use case that employs NS scale out. However, the methodologies for manual scaling and on-demand scaling are similar. The NS scale out is triggered by an increasing the traffic load generated by the Test PNFs. The scaling procedures are initiated by the NFV Orchestrator and the VNF Manager after they detect the increased traffic load.

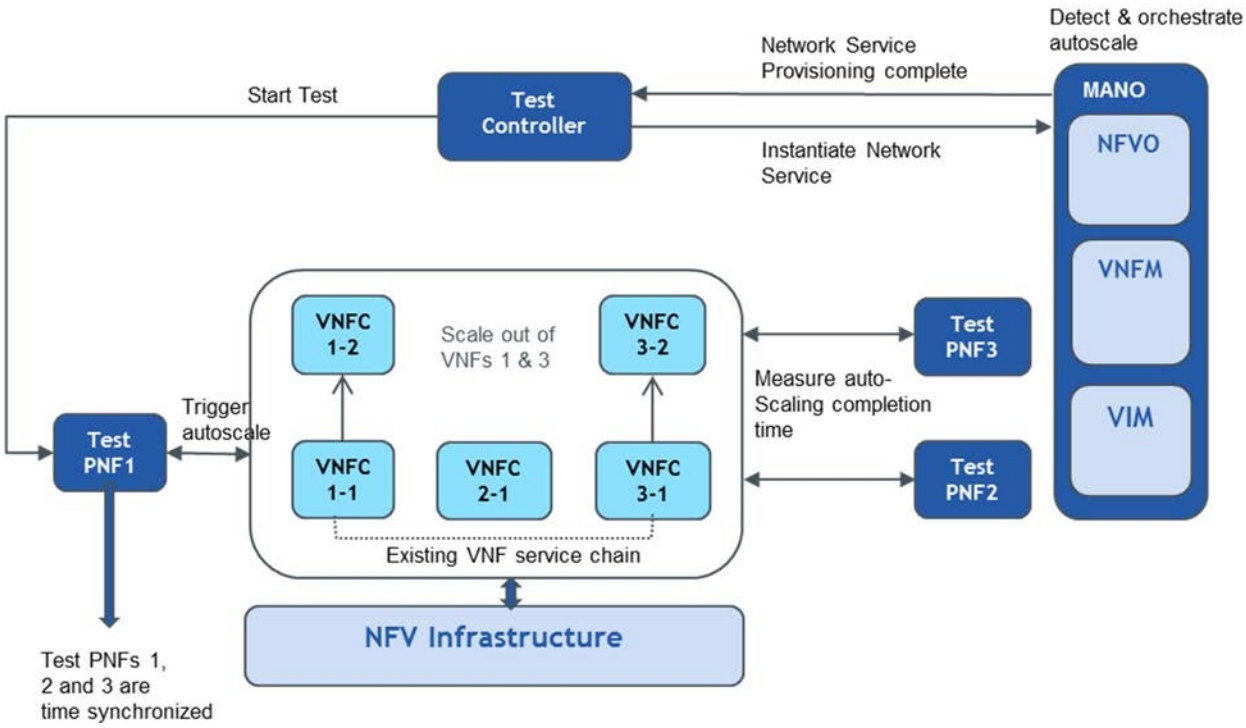

Figure 8.6: Network Services Scaling - Validating autoscaling (scale out)

Network Services Autoscaling validation	
Identifier	NS_Autoscaling_Test_1
Test Purpose	To verify the successful completion of NS autoscaling in response to autoscale stimuli. The NSUT is not the only functional block that is being tested. The MANO components such as the NFV Orchestrator, VNF Manager and the VIM play an active role in the test and are responsible for processing the autoscale stimuli and instantiating VNF components. In effect, the MANO's ability to perform its role in NS autoscaling is also tested. A non-goal of this test is the validation of the MANO components in isolation or the validation of the interfaces between the NSUT and the VNF Manager/NFV Orchestrator.
Configuration	See figures 8.3, 8.4,8.5, and 8.6. The Test PNFs 1 and 2 are connected to the NSUT and they exchange bi-directional network traffic with each other.
References	ETSI GS NFV-MAN 001 [i.8]. ETSI GS NFV-SWA 001 [i.1].
Applicability	This methodology is applicable to a broad range of Network Services and is agnostic to the specific function of the NS or its constituent VNFs.
Pre-test conditions	• The autoscaling capabilities of the individual VNFs that constitute the NS have been fully validated per the methods described in clause 7.1.4 "VNF Scaling". • The user has defined multiple flavours for the NSUT. The test starts with Flavour A of the NSUT. The user has defined the performance target levels for both Flavours A and B of the NSUT. • The NSUT has been assigned the necessary NFVI resources to perform at its performance target for Flavour A. • The Test PNFs 1and 2 have the needed resources and capabilities to exchange control and user plane traffic at the performance target levels of the NSUT Flavour A. • Test Controller is able to access the NSD and access its fields related to autoscale policy and the stimuli that are needed to cause the autoscaling. • The Test PNFs 1, 2 and 3 have the needed resources and capabilities to stimulate the NSUT to scale out to Flavour B (increased traffic or CPU load) • The values of Tmax, the maximum allowed time for NS scale out is known.

Network Services Autoscaling validation				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Test Controller instructs the Test PNFs to initiate the test. Test PNFs 1 and 2 establish the necessary connections with the Network Service endpoints, prior to exchanging traffic with the NSUT Flavour A.	
	2	Check	Validate that the necessary user plane or control plane connections between Test PNFs and NSUT have been established.	
	3	Stimulus	The Test PNFs 1 & 2 originate bi-directional traffic toward the NSUT Flavour A at its performance target level.	
	4	Check	The Test PNFs 1 and 2 exchange bi-directional traffic for at least 10 seconds.	
	5	Check	The Test PNFs ensure that the NSUT Flavour A correctly processes and forwards all packets without errors and ensures that the performance 'P' of the NSUT meets or exceeds its performance target.	
	6	Stimulus	The Test PNFs dial up the traffic and load toward the NSUT Flavour A, to a level that will trigger a scale out to Flavour B. The exact details of the increase in traffic and load are NSUT dependent and are outside the scope of this test methodology. The time T1, when the traffic and load reach the level that will trigger autoscaling is initiated, is noted.	
	7	Stimulus	Starting at time T1, assess the performance 'P' of the NSUT periodically by making traffic measurements at Test PNFs 2 and 3. The exact metrics are NSUT dependent. The polling interval for the NSUT performance measurements are user defined but it is recommended the polling is done once every second. The polling is done for a user defined maximum of Tmax seconds.	
	8	Check	At the end of every polling interval make the following checks at Test PNFs 2 and 3: • Measure the first instance time t = T2, when traffic is observed on PNF3. Log the value of T2. Do not repeat this step after the T2 has been logged the first time. • Compare the measured performance 'P' of the NSUT to the performance target for NSUT Flavour B. If 'P' is lower than the performance target and time t < Tmax, go back to Step 7 and continue polling. • Measure the received traffic at Test PNFs 2 and 3 and ensure that the distribution (load balancing) of traffic across PNF2 and 3 meets the user expectations. If the load balancing is improper and time t < Tmax, go back to Step 7 and continue polling. • If time t > Tmax, go to Test Verdict step. • Else, log time t = T3, as time at which scale out is completed and [T3-T1] as the time needed to complete scale out.	
Test Verdict	The scale out of VNFUT from Flavour A to Flavour B is deemed as a failure if the [T3-T1] > Tmax. The scale out of NSUT from Flavour A to Flavour B is deemed successful if all checks in Step 8 are successful and the time needed to complete scale out is T3 - T1. In addition, present the following metrics to the user: • The value of T1 and T2. • Detailed performance metrics for every polling interval, in the form of a chart during the transition from T1 to T3, highlighting any NS performance degradation below NS Flavour A performance target levels.			

Annex A (informative): Authors & contributors

The following people have contributed to the present document:

Rapporteur:

Mr Rajesh Rajamani (Spirent Communications)

Other contributors:

Mr Pierre Lynch (Ixia)

Mr Jörg Aelken (Ericsson)

Mr Al Morton (AT&T)

Mr Diego Lozano de Fournas (Spirent Communications)

Marie-Paule Odini (Hewlett-Packard)

Mr. Francisco Javier Ramon Salguero (Telefonica)

Annex B (informative): Change History

Date	Version	Information about changes
	0.1.0	Initial version of Final draft for WG approval after approval of chapter 8 for Network Services
	0.1.1	Incorporated feedback from Jersey City plenary and Al Morton's and Pierre Lynch's changes in Chapter 5
12/05/2015	0.1.2	Incorporated Bruno Chatras' suggested changes for formatting, numbering, clause headings and the removal of 'shall' and 'must'
12/11/2015	0.1.3	Incorporated additional feedback from TST WG meeting
01/16/2016	0.1.4	Incorporated changes suggested by Silvia Almagia and Joerg Aelken regarding Editor's notes, references, abbreviations, and other formatting issues
1/17/2016	0.1.5	Removed 7 occurrences of must and replaced with should, per Bruno Chatras' feedback

History

Document history		
V1.1.1	April 2016	Publication