SEVENTH FRAMEWORK PROGRAMME

TNOVA

NETWORK FUNCTIONS AS-A-SERVICE
OVER VIRTUALISED INFRASTRUCTURES

GRANT AGREEMENT NO. 619520

Deliverable D2.51

Planning of trials and evaluation - Interim

Editor G. Xilouris (NCSRD)

Contributors E. Trouva (NCSRD), E. Markakis, G. Alexiou (TEIC), P.Comi, P. Paglierani (ITALTEL), J. Ferrer Riera (i2CAT), D. Christofy, G. Dimosthenous (PTL), J. Carapinha (PTINS), P. Harsh (ZHAW), Z. Bozakov, D. Dietrich, P. Papadimitrioy (LUH), G. Gardikis (SPH).

Version 1.0

Date Dec 20th, 2014

Distribution PUBLIC (PU)

Executive Summary

The validation, assessment and demonstration of the T-NOVA architecture as a complete end-to-end VNFaaS platform, is critical for the success of T-NOVA as an Integrating Project. The aim is not only to present technical advances in individual components, but –mainly- to demonstrate the added value of the integrated T-NOVA system as a whole. To this end, the overall plan for the validation and assessment of the T-NOVA system, to take place in WP7, is mostly concentrated on end-to-end system-wide use cases.

The first step is the assembly of a testing toolbox, taking into account standards and trends in benchmarking methodology as well as industry-based platforms and tools for testing of network infrastructures. Another valuable input is the current set of guidelines drafted by ETSI for NFV performance benchmarking.

The next step is the definition of the overall T-NOVA evaluation strategy. The challenges in NFV environment validation are first identified; namely a) the functional and performance testing of VNFs, b) the reliability of the network service, c) the portability and stability of NFV environments, as well as d) the monitoring of the virtual network service. Then, a set of evaluation metrics are proposed, including system-level metrics (with focus of the physical system e.g. VM deployment/scaling/migration delay, data plane performance, isolation etc.) as well as service-level metrics (with focus on the network service e.g. service setup time, re-configuration delay, network service performance).

The specification of the experimental infrastructure is another necessary step in the validation planning. A reference pilot architecture is defined, comprising NFVI-PoPs with compute and storage resources, each one controlled by the VIM. NFVI-PoPs are interconnected over an (emulated) WAN (Transport Network), while overall management units (Orchestration and Marketplace) interface with the entire infrastructure. This reference architecture will be instantiated (with specific variations) in three integrated pilots (in Athens/Heraklion, Aveiro and Hannover, supported by NCSRD/TEIC, PTIN and LUH respectively), which will assess and showcase the entire set of T-NOVA system features. Other labs participating in the evaluation procedure (Milan/ITALTEL, Dublin/INTEL, Zurich/ZHAW and Limassol/PTIN) will focus on testing specific components/functionalities.

The validation plan is further refined by recalling the system use cases defined in D2.1 and specifying a step-by-step methodology –including pre-conditions and test procedure- for validating each of them. Apart from verifying the expected functional behaviour via well-defined fit criteria, a set of non-functional (performance) metrics, both system- and service-level is defined, for assessing the system behaviour under each UC. This constitutes a detailed plan for end-to-end validation of all system use cases, while at the same time measuring and assessing the efficiency and effectiveness of the T-NOVA architecture.

Last, in addition to use-case-oriented testing, a plan is drafted for testing each of the four VNFs developed in the project (vSBC, vDPI, vSA, vHG). For each VNF, specific measurement tools are selected, mostly involving L3-L7 traffic generators, producing

application-specific traffic patterns for feeding the VNFs. A set of test procedures is then described, defining the tools and parameters to be adjusted during test, as well as the metrics to be collected.

The experimentation/validation plan laid out in the present document will be subject to continuous elaboration throughout the project, depending on the progress of implementation, on the evolution of the technical architecture and the possible adjustment of technology selections.

Table of Contents

Index of Figures

Index of Tables

1. INTRODUCTION

The aim of T-NOVA project is to design and develop an integrated end-to-end architecture for NFV services, covering all layers of the technical framework, from the Marketplace down to the Infrastructure (NFVI). The purpose is to present a complete functional solution, which can be elevated to pre-operational status with minimal additional development after project end.

In this context, the validation, assessment and demonstration of the T-NOVA solution on end-to-end basis becomes critical for the success of T-NOVA as an Integrating Project. The aim is not only to present technical advances in individual components, but –mainly- to demonstrate the added value of the integrated T-NOVA architecture as a whole. To this end, the overall plan for the validation and assessment of the T-NOVA system, to take place in WP7, is mostly concentrated on end-to-end system-wide use cases, rather than on unit tests of individual components or sub-components, which is expected to take place within the respective implementation WP (WP3-WP6).

The present deliverable is a first approach –to be further elaborated in D2.52- to the planning of the validation/experimentation campaign of T-NOVA, describing the assets to be involved, the tools to be used and the followed methodology. Chapter 2 overviews in high-level the overall validation and evaluation methodology framework, highlighting some generic frameworks and recommendations for testing network and IT infrastructures. Chapter 3 discusses the challenges associated with NFV environment validation and identified candidate system- and service-level metrics. Chapter 4 describes the pilot infrastructures (on which the entire T-NOVA system will be deployed) as well as the testbeds, which will be used for focused experimentation. Chapter 5 defines the validation procedures (steps, metrics and fit criteria) to be used for validating each of the T-NOVA Use Cases. Moreover, the procedures for assessing VNF-specific scenarios are described. Finally, Chapter 6 concludes the document.

2. Overall Validation and Evaluation Methodology Framework

This section attempts a survey of the related standard and industry base methodologies available as well as recommendations from ETSI NFV ISG.

2.1. Standards- Based Methodologies Review

2.1.1. IETF

In the frame of IETF, the Benchmarking Methodology WG (bmwg) [BMWG] is devoted to proposing the necessary metrologies and performance metrics to be measured in a lab environment, so that will closely relate to actual observed performance on production networks.

The bmwg WG is examining performance and robustness across various metrics that can be used for validating a variety of applications, networks and services. The main metrics that have been identified are:

- Throughput (min, max, average, standard deviation)
- Transaction rates (successful/failed)
- Application response times
- Number of concurrent flows supported
- Unidirectional packet latency

The group has proposed benchmarking methodologies for various types of interconnect devices. Although these tests are focused on physical devices, the main methodologies might as well be applied in virtualised environments for performance and benchmarking of VNFs. The most relative identified RFCs are:

- RFC 1944 Benchmarking Methodology for Network Interconnect Devices [RFC1944]
- RFC 2889 Benchmarking Methodology for LAN Switching Devices [RFC2889]
- RFC 3511 Benchmarking Methodology for Firewall Performance [RFC3511]

Additionally, the IETF IP Performance Metrics (ippm) WG [IPPM] has released a series of RFCs, related to standard metrics that can be applied to measure the quality, performance, and reliability of Internet data delivery services and applications running over IP. Related RFCs are:

- RFC 2679 A One-way Delay Metric for IPPM [RFC2679]
- RFC 2680 A One-way Delay Metric for IPPM [RFC2680]
- RFC 2681 A Round-trip Delay Metric for IPPM [RFC2681]
- RFC 2498 IPPM Metrics for Measuring Connectivity [RFC2498]

2.1.2. ETSI NFV ISG

The ETSI NFV ISG has proposed in the recent draft on NFV performance (ETSI GS NFV-PER 001 V1.1.1) methodologies for the testing of VNFs [NFVPERF]. The aim is to unify the testing and benchmarking of various heterogeneous VNFs under a common methodology. For the sake of performance analysis, the following workload types are distinguished:

- Data-plane workloads, which cover all tasks related to packet handling in an end-to-end communication between edge applications
- Control-plane workloads, which cover any other communication between NFs that is not directly related to the end-to-end data communication between edge applications.
- Signal processing workloads, which cover all NF tasks related to digital processing such as the FFT decoding and encoding in a C-RAN Base Band Unit (BBU).
- Store workloads, which cover all tasks related to disk storage.

The taxonomy of the workload characterisation is illustrated in Figure 1.

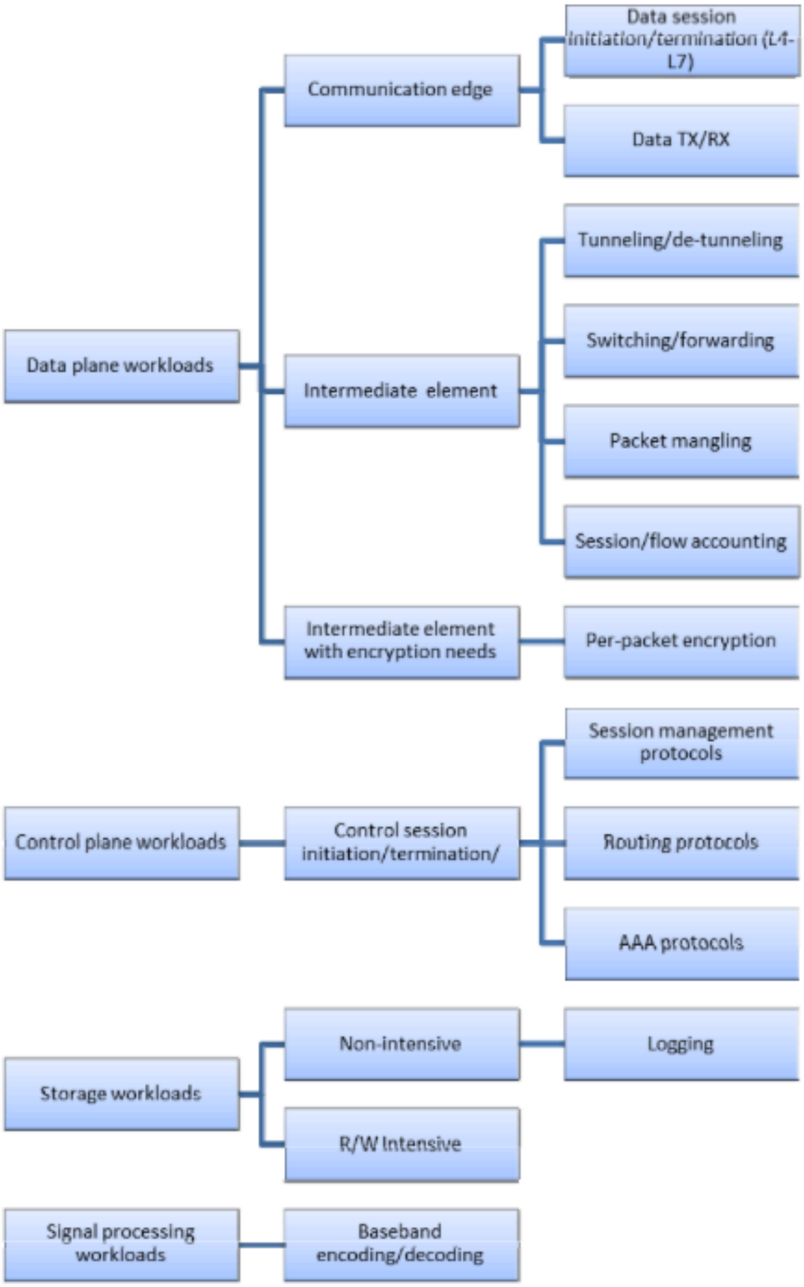

Figure 1 Performance testing workload taxonomy

A mapping of the above taxonomy to the VNFs offered by T-NOVA as a proof of concept is presented in Section **Error! Reference source not found.**.

2.2. Industry benchmarking solutions

For the testing and validation of networks and network application several vendors have developed solutions for automatic stress testing with a variety of network technologies and protocols ranging from L2 to L7. Among these the most prominent are IXIA [IXIA], and Spirent [SPIRENT]. They both adopt standardised methodologies, benchmarks and metrics for the performance evaluation and validation of a variety of physical systems. Lately, due to the ever increasing need for testing in the frame of NFV, they have also developed methodologies that address the need for benchmarking in virtualised environments.

2.2.1. Spirent

Spirent supports standards-based methodologies for the NFV validation. In general, the methodologies used are similar to those employed to physical Devices Under Test (DUT). The functionalities and protocols offered by standard hardware devices, also have to be validated in a virtual environment. VNF performance is tested against various data plane and control plane metrics, including:

- Data plane metrics:
  - latency;
  - throughput and forwarding rate;
  - packet-delay variation and short-term average latency;
  - dropped and errored frames.
- Control plane metrics:
  - States and state transitions for various control plane protocols;
  - Control plane frames sent and received on each session;
  - Control plane error notifications;
  - Validation of control-plane protocols at high scale;
  - Scaling up on one protocol and validating protocol state machines and data plane;
  - Scaling up on multiple protocols at the same time and validating protocol states machines and data plane;
  - Scaling up on routes and MPLS tunnels. These are a representative sample of a comprehensive set of control-plane and data-plane statistics, states and error conditions that are measured for a thorough validation of NFV functions.

2.2.2. IXIA

Ixia’s BreakingPoint Resiliency Score [IXIABRC] and the Data Center Resiliency Score are setting standards against which network performance and security (physical or virtual) can be measured. Each score provides an automated, standardized, and deterministic method for evaluating and ensuring resiliency and performance.

The Resiliency Score is calculated using standards by organizations such as US–CERT, IEEE, and IETF, as well as real-world traffic mixes from the world’s largest service providers. Users simply select the network or device for evaluation and the speed at which it is required to perform. The solution then runs a battery of simulations using a blended mix of application traffic and malicious attacks. The Resiliency Score simulation provides a common network configuration for all devices in order to maintain fairness and consistency for all vendors and their solutions.

The Resiliency Score is presented as a numeric grade from 1 to 100. Networks and devices may receive no score if they fail to pass traffic at any point or they degrade to an unacceptable performance level. The Data Center Resiliency Score is presented as a numeric grade reflecting how many typical concurrent users a data center can support without degrading to an unacceptable quality of experience (QoE) level. Both scores allow quick understanding of the degree to which infrastructure performance, security, and stability will be impacted by user load, new configurations, and the latest security attacks.

By using the Resiliency Score, it is possible to :

- Measure the performance of Virtual Network Functions (VNFs) and compare it to its physical counterparts;
- Measure the effect of changes to virtual resources (VMs, vCPUs, memory, disk and I/O) on VNF performance, allowing to fine tune the virtual infrastructure to ensure maximum performance;
- Definitively measure the number of concurrent users which a virtualized server will support before response time and stability degrade;
- Measure application performance in physical and virtual environments.

3. T-NOVA Evaluation Aspects

This chapter provides a preliminary description of the T-NOVA evaluation aspects from the architectural and functional perspective. These aspects will be used for the definition of the evaluation strategy and as a starting point for the validation activities within WorkPackage 7.

3.1. Challenges in NFV Environment Validation

This section provides an overview of the challenges involved in the validation procedures for NFV environments.

Functional and performance testing of network functions - In the general case where the performance testing results are provided for end-user consumed network services, the primary concern is their application performance and the exhibited quality of experience. The view in this case is more macroscopic and does not delve to the protocol level or to the operation of e.g. BGP, routing or CDN functionalities. However for the Operators, additional concerns exist, regarding specific control plane and data plane behaviour; whether, for example the number of PPPoE sessions, throughput and forwarding rates, number of MPLS tunnels and routes supported are broadly similar between physical and virtual environments. Testing must ensure that the performance of virtual environments is equivalent to that of the corresponding physical environment and provide the appropriate quantified metric to support it.

Validating reliability of network service - Operators and users are accustomed to 99.999 percent availability of physical network services and will have the same expectations for virtual environments. It is important to ensure that node, link and service failures are detected within milliseconds and that corrective action is taken promptly without degradation of services. In the event that virtual machines are migrated between servers, it is important to ensure that any loss of packets or services is within acceptable limits set by the relevant SLAs.

Ensuring portability of VMs and stability of NFV environments - The ability to load and run virtual functions in a variety of hypervisor and server environments must also be tested. Unlike physical environments, instantiating or deleting VMs can affect the performance of existing VMs as well as services on the server. In accordance with established policies, new VMs should be assigned the appropriate number of compute cores and storage without degrading existing services. It is also critically important to test the virtual environment (i.e. NFVI and VNFs), including the orchestrator and Virtual Infrastructure Management (VIM) system.

Active and passive monitoring of virtual networks - In addition to pre-deployment and turn-up testing, it is also important to monitor services and network functions on either an on-going, passive basis or an as-needed, active basis. Monitoring virtual environments is more complex than their physical equivalents because operators need to tap into either an entire service chain or just a subset of that service chain. For active monitoring, a connection between the monitoring end-points must also be created on an on-demand basis, again without degrading the performance of other functions that are not being monitored in that environment.

3.2. Definition of relevant metrics

In the context of VNF-based services, validation objectives should be defined based not only on traditional service performance metrics, which are generally applicable to network services (e.g. data plane performance – maximum delay, jitter, bit error rate, guaranteed bandwidth, etc), but also new NFV-specific metrics related to resource automated provisioning and multi-tenancy – e.g. time to deploy a new VNF instance, time to scale out/in, isolation between tenants, etc. On the other hand, validation objectives should be defined both from system and service perspectives, which are considered separately in the following sub-sections.

3.2.1. System level metrics

The system level metrics address the performance of the system and its several parts, without associating to a specific NFV service. The following is a preliminary list of system level metrics to be checked for validation purposes. Although the overall system behaviour (e.g., performance, availability, security, etc.) depends on the several sub-systems or component, for evaluation purposes we are only interested in service high-level goals and the performance of the system as a whole.

- Delay related metrics:
    - Time to deploy a VM
    - Time to scale-out a VM
    - Time to scale-in a VM
    - Time to migrate a VM
    - Time to establish a virtual network
    - Time to map a service request into the physical infrastructure
- Data plane performance:
    - Maximum achievable throughput between any two points in the network
    - Packet delay (between any two points in the network)
- Performance under transient conditions
    - Stall under transient conditions (e.g. VM migration, VM scale-out/in)
    - Time to modify an existing virtual network (e.g. insertion of new node, reconfiguration of topology)
- Isolation in multi-tenant environment
    - Variability of data plane performance with the number of tenants sharing the same infrastructure resource
    - Variability of control plane performance with the number of tenants sharing the same infrastructure resources

3.2.2. Service level metrics

Service level metrics are supposed to reflect the service quality experienced by end users. Often, this kind of metrics is used as the basis for SLA contracts between service providers and their customers.

In general, NFV services may have different levels of complexity and service level objectives may vary as a result of that variability. On the other hand, different types of NFV services may have different degrees of sensitiveness to impairments.

- Time related metrics
  - Time to start a new VNF instance (interval between submission request through the customer portal and the time when the VNF becomes up and running).
  - Time to modify/reconfigure a running VNF (interval between submission of the reconfiguration request through the customer portal and the time when the modification is enforced).
- Data plane performance
  - Maximum achievable throughput in a customer virtual network
  - Latency (packet delay) between any two points in the customer virtual network
- Performance under transient conditions
  - Impact of inserting / removing a VNF / VNF chain in the data path on the network connectivity service already in place (transient packet loss)
  - Impact of inserting / removing a VNF / VNF chain in the data path on the end-to-end delay
  - Impact of inserting / removing a VNF / VNF chain in the data path on data throughput

4. Pilots and Testbeds

This chapter contains the description of the different test-beds involved in the T-NOVA project, as well as the description of the different pilots, which will be used to perform all the testing and validation activities.

4.1. Reference Pilot architecture

4.2. T-NOVA Pilots

In order to guide the integration activities, a reference pilot architecture is elaborated. A preliminary view of the reference architecture is illustrated in Figure 2. It corresponds to a complete implementation of the T-NOVA system architecture described in D2.21, including a single instance of the Orchestration and Marketplace layers, one or more NFVI-PoPs, each one managed by a VIM instanced, interconnected by a (real or emulated) WAN infrastructure (core, edge and access).

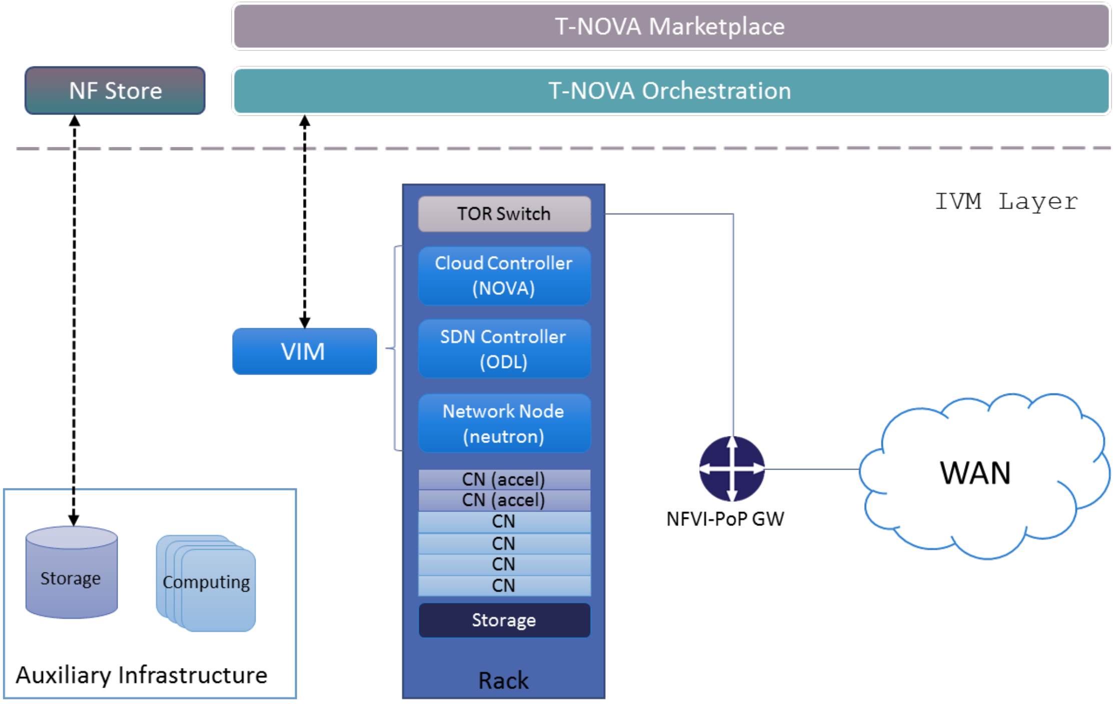

Figure 2 T-NOVA Pilot reference architecture

The reference pilot architecture will be enriched as the T-NOVA implementations progress, and will be detailed and refined in order to present finally all the building blocks and components of the Pilot deployment. The architecture will be implemented in several pilot deployments, as detailed in the next section. However, in each pilot deployment, given the equipment availability and the specific requirements for Use Case validation, the reference architecture will be adapted appropriately. Starting the description from bottom up, the Infrastructure Virtualisation and Management Layer includes:

- an execution environment that provides IT resources (computing, memory and storage) for the VNFs. This environment comprises i) Compute Nodes (CNs) based on x86 architecture commodity hardware without particular platform capabilities and ii) enhanced CNs (eCNs) that are similarly based on x86 architecture commodity hardware enhanced with particular data processing acceleration capabilities (i.e. DPDK, AES-NI, GPU acceleration).
- a Cloud Controller node (one per NFVI-PoP) for the management and control of the aforementioned IT resources, based on Openstack platform
- a Network Node (one per NFVI-PoP), running OpenStack Neutron service for managing the in-cloud networking created by OpenVirtualSwitch instance in each CN and also in the Network Node.
- an SDN Controller (one per NFVI-PoP), based on the recent version of OpenDayLight platform, for the control of the virtualised network resources. The interaction and integration of the SDN controller with the OpenStack platform is achieved via the ML2 Plugin component provided by Neutron service.

The latter three components along with the implemented interfaces and agents belong to the Virtualisation Infrastructure Management block (along with other VIM components –not fully detailed) as illustrated in more detail in Figure 3.

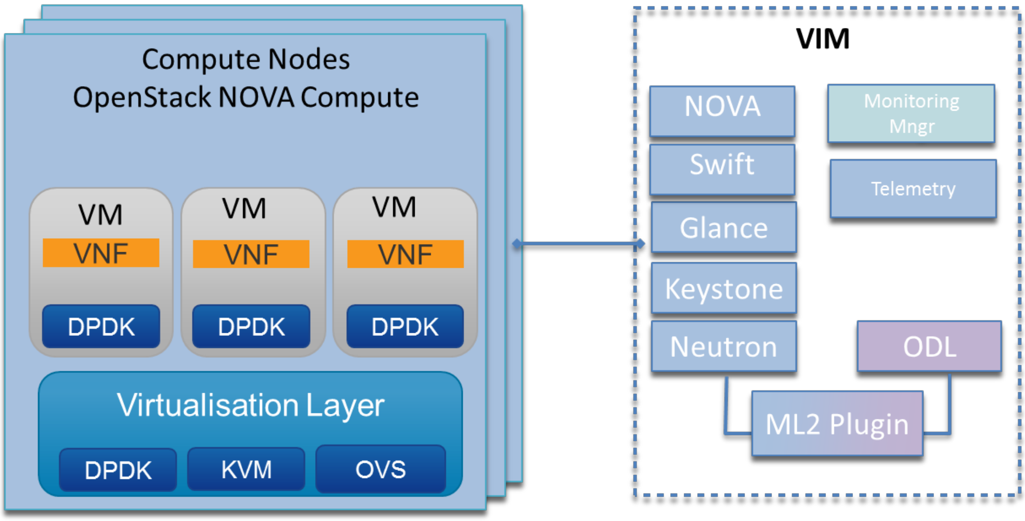

Figure 3 VIM and Compute Node details

It is anticipated that the integration of the ODL with the Openstack in-cloud network controller (Neutron) is achieved via the ML2 plugin. In this sense Openstack is able to control the DC network through this plugin and having ODL control OVS instances via OpenFlow protocol. However in order to provide access to specific to T-NOVA functionalities, the VIM will allow immediate orchestration communication with the ODL controller. Please refer to deliverables D2.31 and D4.01 for more details on the VIM components and structure.

The connectivity of this infrastructure with other deployed NFVI-PoP it is realized via a L3 gateway. As it can be observed, in addition to NFVI-PoP equipment it is anticipated that an auxiliary infrastructure exists to facilitate the deployment of centralised components, such as the Orchestrator and the Marketplace modules.

4.2.1. Athens-Heraklion Pilot

4.2.1.1. Infrastructure and topology

The Athens-Heraklion pilot will be based on a distributed infrastructure between Athens (NCSRD premises) and Heraklion (TEIC premises). The interconnection will be provided by the Greek NREN (GRNET). This facility is freely available for the academic institutes, supporting certain levels of QoS. The idea behind this Pilot is to be able to demonstrate T-NOVA capabilities over a distributed topology with at least two NFVI-PoPs, interconnected by pre-configured links. The setup is ideal for experimentation with NS and VNF deployment issues, and performance taking into account possible delays and losses in the interconnecting links. Additionally, this Pilot will offer to the rest of the WPs a continuous integration environment in order to allow verification and validation of the proper operation of all developed and integrated software modules.

(a) Athens infrastructure

The Pilot architecture that will be deployed over NCSRD testbed infrastructure is illustrated in Figure 4.

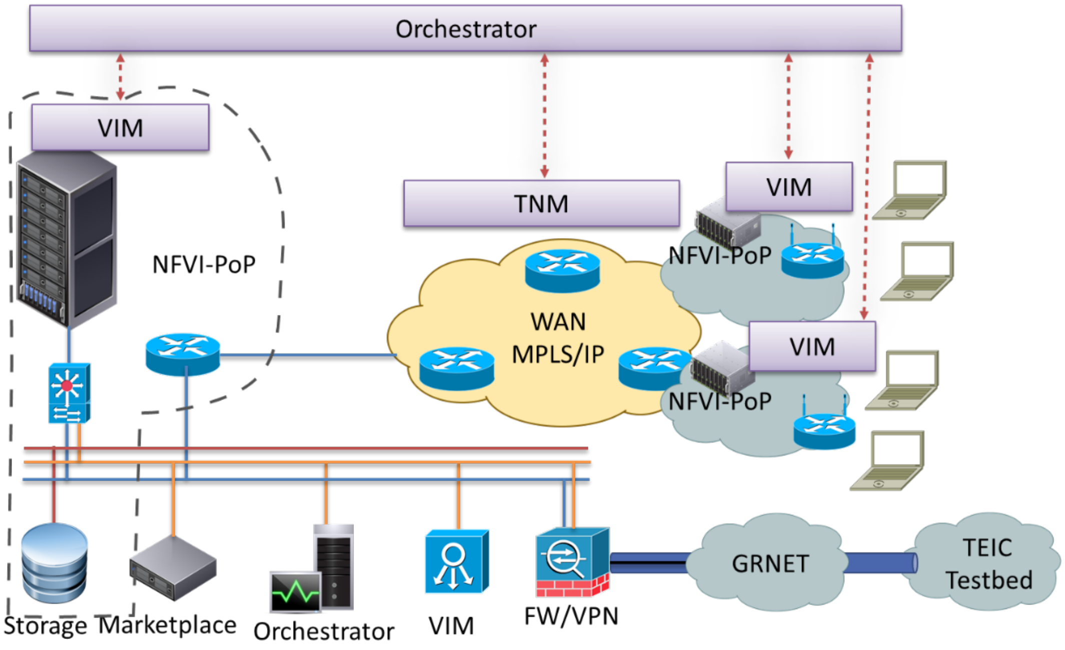

Figure 4 Athens topology

The detailed specifications of Athens infrastructure are summarised in the following tables (Table 1,)

Main NFVI-PoP

Table 1 Main NFVI-PoP Specifications

OpenStack Controler	Server Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 4cores, 16GB RAM, 1TB storage, Gigabit NIC
Openstack Compute	2 Servers Each with 2x (Intel(R) Xeon(R) CPU E5620@2.40GHz, 4cores), 56GB RAM, 1TB storage, gigabit NICs.
Openstack Network Node **(neutron)**	Server Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz 8GB RAM (to be upgraded)
OpenDaylight	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz 8GB RAM (to be upgraded)
Storage	8TB, SCSI, NFS NAS
Hypervisors	KVM
Cloud Platform	Openstack Juno
Networking	PICA8 Openflow 1.4 switch

Table 2 Edge NFVI-PoP Specifications

OpenStack All–in-one, plus ODL	Server Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 4cores, 16GB RAM, 2TB storage, Gigabit NIC
Storage	8TB, SCSI, NFS NAS
Hypervisors	KVM
Cloud Platform	Openstack Juno
Networking	PICA8 Openflow 1.4 switch

In addition to a full-blown deployment of an NFVI-PoP (backbone DC), the accommodation of a legacy network domain (non-SDN) is also considered in the pilot architecture. This network domain will act as Transport network, providing connectivity to other more simple NFVI-PoPs. These PoPs will be deployed using an all-in-one logic, where the actual IT resources are implemented around a single commodity server (with limited of course capabilities). However, the selection of the above topology is justified by the need to be able to validate and evaluate the Service Mapping components and experiment with VNF scaling scenarios. NCSRD and TEIC infrastructure being already interconnected via the Greek NREN, which is GRNET, it is fairly easy to be interconnected and constitute a distributed Pilot for T-NOVA experimentation. This will provide the opportunity to evaluate NS composition and Service Function Chaining issues over a larger than a laboratory testbed deployment over 100% controllable conditions (depending on the SLA with our NREN).

(b) TEIC Infrastructure

In TEIC premises a full implementation of the T-NOVA testbed will be deployed conforming to the reference pilot architecture as described in this deliverable. The IT infrastructure that will be used for T-NOVA experimentation and validation is detailed in the following table (Table 3:

Table 3 TEIC IT Infrastructure description

Servers	8 (10 cores, CentOS, 1TB Hard Disk, RAM 36GB)
CPUs	80 cores
RAM	280GB
Storage	8TB
Hypervisors	KVM
Cloud Platform	Openstack Juno
Networking	PICA8 Openflow 1.4 switch

The PASIPHAE Lab of TEIC features of various access network (Table 4) that can be used if needed to emulate the access part of the T-NOVA network in large scale deployment

Table 4 TEIC Access Network Description

DVB-T Network	DVB-T Network (100 real users)
WiMAX	WiMAX Network (100 real users)
Ethernet	Local laboratory equipped with 300 PCs
WiFi	Campus WiFi Infrastructure with 1000 users

4.2.1.2. Deployment of T-NOVA components

TEIC plans to have a full T-NOVA deployment (i.e. including all the T-NOVA stack components) to be able to run local testing campaigns but also participate to distributed evaluation campaigns along with federated Athens Pilot.

4.2.2. Aveiro Pilot

4.2.2.1. Infrastructure and topology

PTInS' testbed facility is targeted at experimentation in the fields of Cloud Networking, network virtualization and SDN. It distributed across two sites, PTInS headquarters and Institute of Telecommunications (IT), both located in Aveiro, as shown in the figure below (Figure 5). The infrastructure includes Openstack-based IT virtualized environments, an OpenDaylight-controlled OpenFlow testbed and a legacy IP/MPLS network domain based on Cisco equipment (7200, 3700, 2800). This facility has hosted multiple experimentation and demonstration activities, in the scope of internal and collaborative R&D projects.

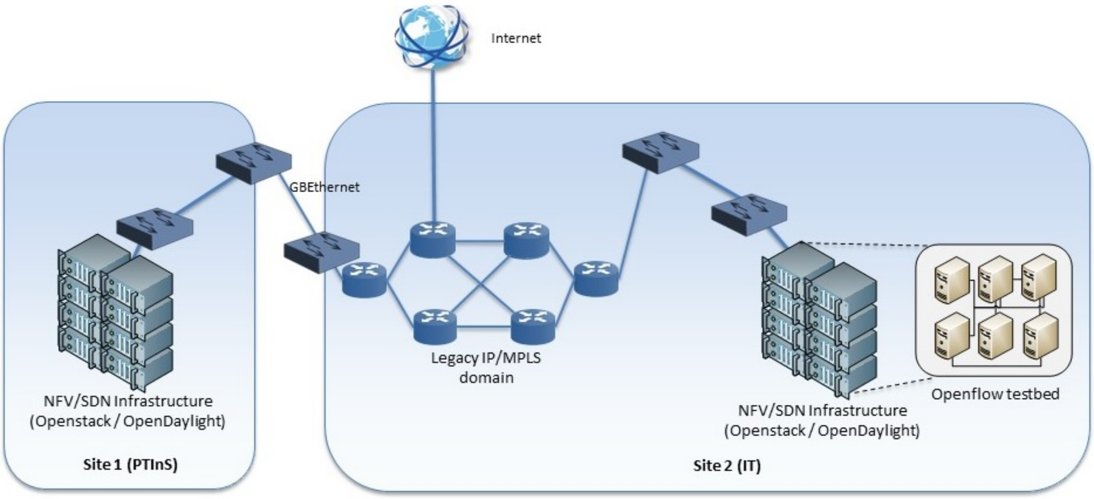
Internet
GBEthernet
Legacy IP/MPLS domain
NFV/SDN Infrastructure (Openstack / OpenDaylight)
Site 1 (PTInS)
NFV/SDN Infrastructure (Openstack / OpenDaylight)
Openflow testbed
Site 2 (IT)

Figure 5 Aveiro Pilot

The infrastructure is as follows:

PTInS:

4 x CPU Xeon E5-2670, 128 GB RAM, 1.8 TB HDD

IT:

- Intel Xeon/ Intel Core i7 cores, currently totaling 157 GB RAM and 40 Cores
- OpenFlow-based infrastructure (4 Network Nodes with OpenvSwitch) controlled by OpenDaylight SDN platform (Hydrogen release)
- IP/MPLS infrastructure (Cisco 7200, 2800, 3700)

4.2.2.2. Deployment of T-NOVA components

PTInS will be able to host all components of the NFV infrastructure. Distributed scenarios involving multiples NFVI PoPs separated by legacy WAN domain will also be easily deployed taking advantage of the IP/MPLS infrastructure available at the lab.

4.2.3. Hannover Pilot

4.2.3.1. Infrastructure and topology

Future Internet Lab (FILab) – illustrated in Figure 6 - is a medium-scale experimental facility owned by the Institute of Communications Technology at LUH. FILab provides a controlled environment in which experiments can be performed on arbitrary, user-defined network topologies, using the Emulab management software.

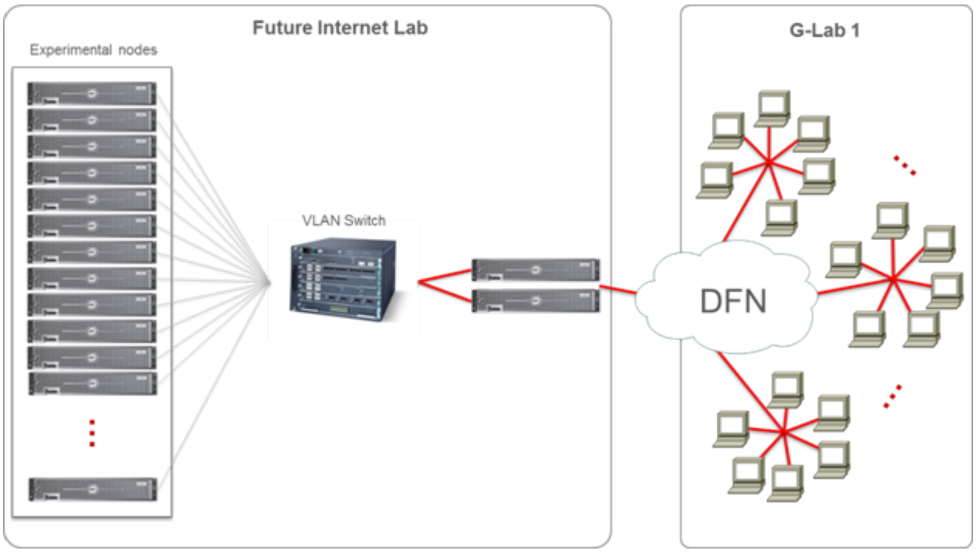

Figure 6 Hannover Pilot architecture

FILab provides an experimental test-bed composed of:

- 60 multi-core servers
    - Intel Xeon E5520 quad-core CPU at 2.26 GHz
    - 8 GB DDR3 RAM at 1333 MHz
    - 1 NIC with 4 x 1G ports
    - Interconnected by a CISCO 6900 switch with 720 Gbps backplane switching capacity and 384 x 1G ports. Each one of these servers is equipped with, 6 GB DDR3 RAM at 1333 MHz and.
- 15 multi-core servers
    - Intel Xeon X5675 six-core CPU at 2.66GHz
    - , 1 NIC with 2 x 10G ports
    - Interconnected by a CISCO NEXUS 5596 switch with 48 x 10G ports. ,
- 22 programmable NetFPGA cards
- 20 wireless nodes, and high-precision packet capture cards
- Various software packages for server virtualization (e.g., Xen, KVM), flow/packet processing (e.g., OpenvSwitch, FlowVisor, Click Modular Router, Snort) and routing control (e.g., NOX, POX, XORP) have been deployed into FILab allowing the development of powerful platforms for NFV and flow processing.

4.2.3.2. Deployment of T-NOVA components

This Pilot will focus on the experimentation and validation of the SDN Control framework to be developed by T-NOVA. Additionally resource mapping mechanisms will also be validated.

4.3. Test-beds for focused experimentation

4.3.1. Milan (ITALTEL)

4.3.1.1. Description

ITALTEL testing labs are composed by interconnected test plants (located in Milan and Palermo, Italy) and based on proprietary or third party equipment, to emulate real-life communication networks and carry out experiments on any type of voice and/or video over IP service. The experimental testbed to verify the behavior of the virtual SBC will be based on the available hardware platforms in Italtel test plants. A simplified scheme representing the connection of two Session Border Controllers is shown in Figure 7

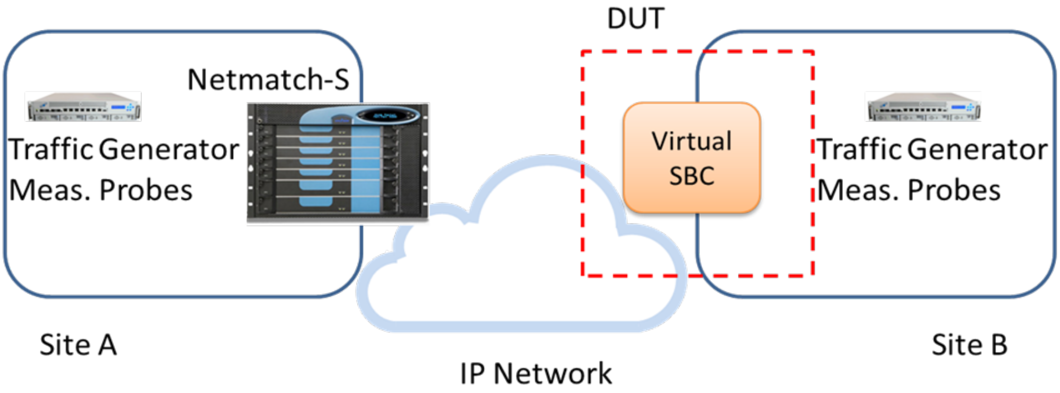

Figure 7 Simplified scheme of Italtel test plant for vSBC characterization

In the scheme, two domains, here referred to as Site A and B, are interconnected through an IP network, and by using two Session Border Controller.

The virtual SBC, which represents the Device under Test, will be connected to Site B. By exploiting the capabilities offered by Italtel test lab, a number of experiment will be designed in order to verify the DUT behavior under a wide variety of test conditions.

The SBC in Site A is the current commercial solution of Italtel, namely the Italtel Netmatch-S. Netwmatch-S is a proprietary SBC, based on bespoke hardware, which can perform a high number of concurrent sessions, and provide various services, such as NAT and Transcoding, both of audio and video sessions. A variety of end-user terminals are present in the test plant, and can be used in order to perform testing on any type of service. In the lab, also High Definition video communication and Tele-presence solutions are present, and can be used for testing activities. Traffic generators are available, to verify the correct behavior of the proposed solutions under loading traffic conditions. Finally, different types of Measurement Probes can be used, which can evaluate different Quality of Service parameters, both intrusively and non-intrusively.

4.3.1.2. Test Planning

The testbed will be mostly used for the validation and the performance optimisation of the SBC VNF.

4.3.2. Dublin (INTEL)

4.3.2.1. Description

The Intel Labs Europe test-bed is medium scale data centre facility comprising of 35+ HP servers of version generations ranging from G4 to G9. The CPUs are XEON based with differing configurations of RAM on-board storage. Additional external storage options in the form of Backblaze storage servers are also available. This heterogeneous infrastructure is available as required by the T-NOVA project. However for the initial experimental protocols a dedicated lab based configuration will be implemented as outlined in Figure 8. This testbed will be dedicated to the initial research activities for Task 3.2 (resource repository) and Task 4.1 (virtualised infrastructure). The nodes will be a mixture of Intel i7 4770, 3,40Ghz CPUs with 32 GB of RAM and one with 2 Xeon E5 2680 v2, 2.8GHz and 64GB of RAM. The latter provides 10 cores per processor (the compute node has in total 20 cores) and provides a set of platform features of interest to Task 4.1 and 3.2 (e.g. VT-x, VT-d, Extended page tables, TSX-NI, Trusted Execution Technology (TXT) and 8GT/s Quick Path Interconnects for fast inter socket communications). Each compute node features an X540-T2 network interface card. The X540 has dual Ethernet 10GB ports which are DPDK-compatible and is SR-IOV capable with support for up to 64 virtual functions. In the testbed configuration one port on the NIC is connected to a Management Network and the other is connected to a Data Network. Inter Virtual Machine traffic on different compute nodes is facilitated via an Extreme Networks G670 48 port SDN switch with OpenFlow support. The management network is implemented with a 10GB 12 port Netgear Pro Safe switch. From a software perspective the testbed is running the IceHouse version of OpenStack and the Hydrogen version of OpenStack. Once the initial configuration has been functionally validated the testbed will be upgrade to Juno and Helium version releases. Integration between OpenStack Neutron module and OpenDaylight is implemented using the ML2 plugin. Virtualisation of the compute resources is based on the use of KVM hypervisors and a libvirt hypervisor controller. DPDK vSwitch delivers virtual VM connectivity through the Data Network.

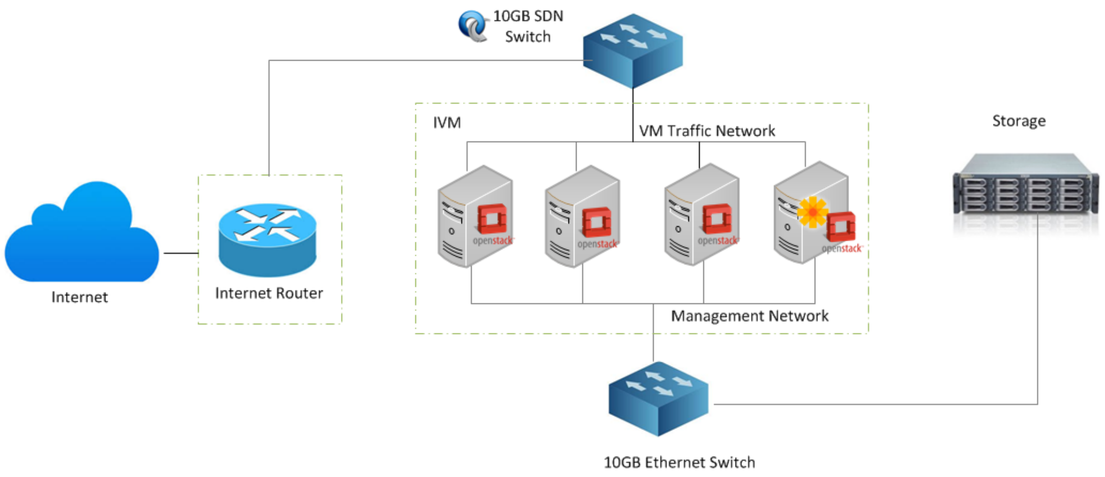

Figure 8 Dublin Testbed

4.3.2.2. Test Planning

An experimental protocol to be executed on the testbed is currently being developed. The initial version of the protocol will be available in deliverable 4.01. The expected activities are based on a number of categories:

- **Workload Characterisation** - Capture of dynamic metrics and identification of metrics which have the most significant influence on workload performance. Identification of opportunities to create derived or synthetic metrics which can be indicative for context based performance.
- **Technology Characterisation** – Evaluate the candidate technologies for the IVM (e.g. vSwitch vs DPDK vSwitch) and identify the most appropriate configurations of implementation and identify any dependencies such software libraries etc.
- **Functional Validation** – Evaluate test-bed behaviour and performance.
- **Enhanced Platform Awareness** - Identify options to implement enhanced platform awareness within the context of the existing capabilities of OpenStack.

4.3.3. Zurich (ZHAW)

Institute of Applied Information Technology (InIT)'s cloud computing lab (ICCLab) at Zurich University of Applied Sciences (ZHAW) run multiple cloud testbeds for research and experimentation purposes. Below are the summary of various experimentation cloud testbeds maintained by the lab (current as of Nov 21, 2014).

Testbed Name	No. of vCPUs	RAM (GB)	Storage (TB)	Purpose
Lisa	200	840	14.5	Used for education and by external community
Bart	64	377	3.2	General R&D projects
Arcus	48	377	2.3	Energy research

XiFi	192	1500	25	Future Internet Zurich Node

4.3.3.1. Description

ICCLab's bart openstack cloud is generally used for various R&D projects. This cloud consists of 1 controller node, and 4 compute nodes, each being Lynx CALLEO 1240 servers. The details of each server is described next.

Type	Lynx CALLEO Application Server 1240
Model	SA1240A304R (1HE)
Processor	2x INTEL® Xeon® E5620 (4cores)
Memory	8x 8GB DD3 SDRAM, 1333MHz, reg. ECC
Disk	4x 1 TB Enterprise SATA-3 Hard Disk, 7200 U/min, 6 Gb (Seagate ST1000NM0011)

Each of the nodes of this testbed is connected through 1 GBps ethernet links to HP ProCurve 2910AL switch, and using 1 GB/s link to ZHAW university network. This testbed has been allocated 32 public IPs in 160.85.4.0/24 block which allows collaborative work to be conducted over this testbed. ICCLab currently has 3 OpenFlow switches that can be provisioned for use in T-Nova at a later point. The characteristics of these switches are:

Model	Pica8 P-3290
Processor	MPC8541
Packet Memory Buffer	4MB
Memory	512MB System / 2GB SD/CF
OS	PicOS, stock version

The schematic of ICCLab's bart testbed (Figure 9) which currently runs OpenStack Havana with VPNaaS, LBaaS enabled is shown in the figure below. Virtualization in each physical node is supported through KVM using libvirt.

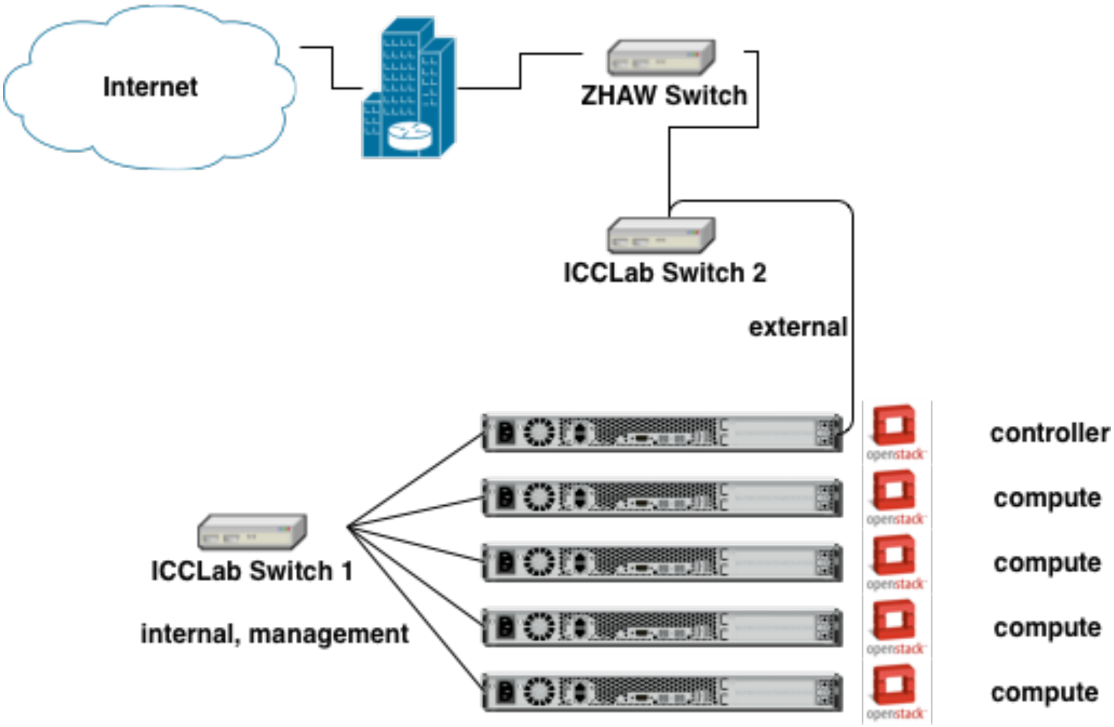

Figure 9 ICCLab "bart" testbed topology

This testbed can be easily modified to add more capacity if needed. Initially this testbed will be used to support ZHAW's development work in T3.4 Service Provisioning, Management and Monitoring, and task 4.3 SDK for SDN. Later, this testbed can be used for T-Nova consortium wide validation tests as a Zurich point-of-presence (POP) (site) for the overall T-Nova demonstrator. For inter-site tests, our testbed can be connected to remote sites through VPN setup.

4.3.3.2. Test Planning

ZHAW testbeds will be used to validate the full T-Nova stack and will be configured as a POP for deploying the NFs through the orchestrator. Furthermore, the SDK for SDN tool that will be developed in T4.3 will undergo functional testing using ZHAW testbed. The exact test planning is still in progress, but the tests will be categorized under three broad categories -

- SDK for SDN functional validation - The set of tests will be planned to undertake the feature coverage and functional evaluation of the SDK for SDN toolkit. For this, the testbed will be modified with addition of OpenFlow switches and SDN controllers
- Testbed validation - The set of tests will be planned to evaluate the general characteristics of the OpenStack testbed itself, VM provisioning latency studies, etc.
- Billing functional validation - The set of tests will be planned together with ATOS to verify the different billing stakeholder scenarios.

4.3.4. Limassol (PTL)

4.3.4.1. Description

PrimeTel’s Triple play platform, called Twister, is a converged end-to-end telecom platform, capable of supporting an integrated multi-play network for various media, services and devices. The platform encompasses all elements of voice, video and data in a highly customisable and upgradeable package. The IPTV streamers receive content from satellite, off-air terrestrial and studios and convert it to MPEG-2/MPEG-4 over UDP multicast, while Video on demand services are delivered over UDP unicast. Twister telephony platform uses Voice over IP (VoIP) technology. The solution is based on open SIP protocol and provides all essential features you expect from Class 5 IP Centrex softswitches. Media Gateways are used for protocol conversion between VoIP and traditional SS7/ISDN telephone networks. IP interconnections with international carriers are provided through international POPs. It also includes components that provide centralized and distributed traffic policy enforcement, monitoring and analytics in an integrated management system. Twister Converged Billing System provides mediation, rating and bill generation for multiple services. It maintains also a profile for each subscriber. The customer premises equipment (CPE) provides to customers Internet, telephony and IPTV connection. It behaves as an integrated ADSL modem, IP router, Ethernet switch and VoIP media gateway. STB receives multicast/unicast MPEG-2/MPEG-4 UDP streams and shows them on TV. Through a Sonus interface and IP Connectivity the platform is linked to partner’s 3G Mobile Network for offering IP services provisioning to mobile customers.

R&D Testbed

PrimeTel’s R&D test-bed facilities can connect to the company’s network backbone and utilize the network accordingly. Through the R&D test-bed research engineers can connects to parts of interest on the real network. In collaboration with the Networks Department R&D could conduct network analysis, traffic monitoring, power measurements etc. and also allow for testing and validation of newly introduced components as part of the its research projects and activities. A number of beta testers could be connected to the test-bed for supporting validation acting as real users and providing the necessary feedback of any proposed system, component or application developed.

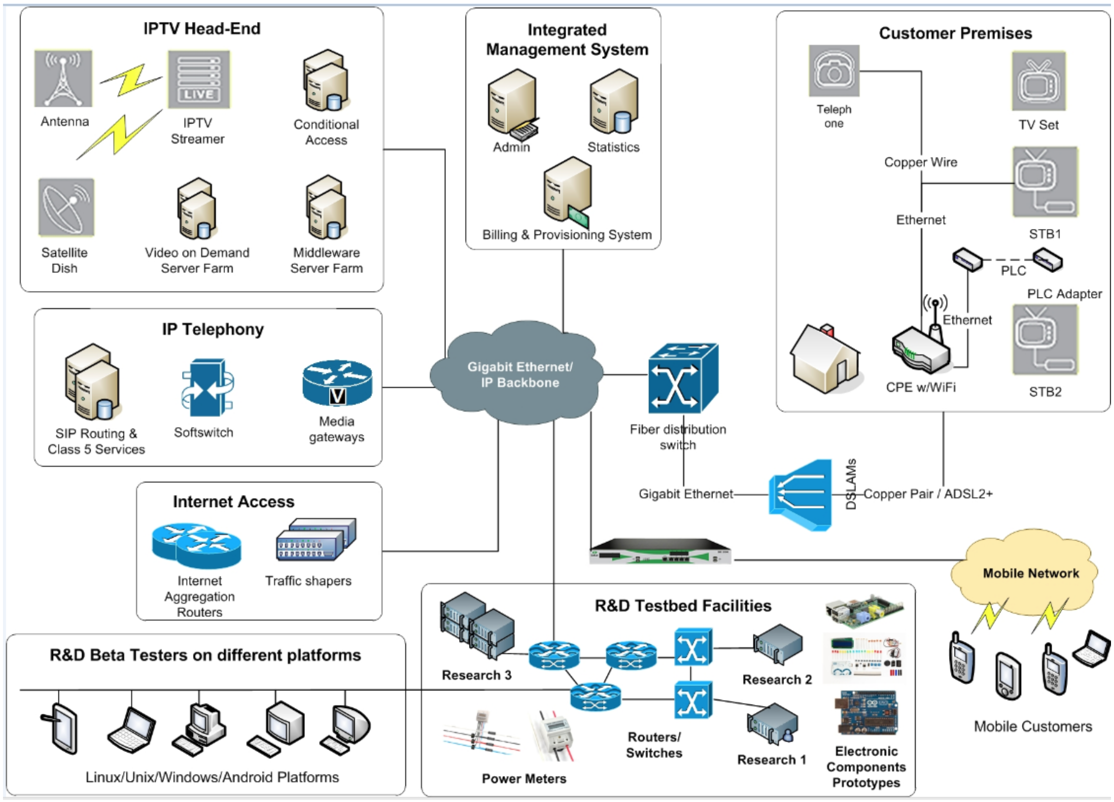

Figure 10 Limassol TestBed

Interconnections

Interconnections with other test-beds could be achieved with VPN tunnels over the Internet.

4.3.4.2. Test Planning

PrimeTel's test-bed is ideal for running the virtual home-box use case and more ideally to test this with real end users. PrimeTel currently has around 12000 TV subscribers, amongst them a number of who have expressed interest in participating in testing and evaluation activities. It is foreseen to allow real end user testing of T-NOVA platform, specifically for testing HG VNF. PrimeTel's Beta Testers (around 100) will be invited to participate in the T-NOVA trials during Y3, more specifically.

5. Trials, Expected Results and Metrics

5.1. System Level Validation

This section approaches the system level validation needs by providing a step-by-step approach for the validation of the T-NOVA Use Cases as they have been laid out in Deliverable D.2.1 [D2.1]. For each UC, the test description includes preconditions, methodology, metrics and expected results. The final, detailed System Validation testing regime will be provided with the second (final) version of this Deliverable.

5.1.1. UC 1 – Compose NFV services

Step Number	1.1
Step Description	The broker requests Service Composition after Customer request, following the submission of the Customer requirements.
Precondition	Customer is authenticated, and is using T-NOVA dashboard for browsing the catalogues and providing his/her requirements
Involved T-NOVA Components	Dashboard, Broker, Orchestrator.
Test Methodology	Automatic service composition request generation in different rates in order to simulate possible real case of Customers requesting composition of particular services
Metrics	Evaluate the latency between the service composition request and the acknowledgement by the Orchestrator. Error rate and misses will also be measured.
Expected Results	The latency should be less that 60 second.

5.1.2. UC1.1 – Browse / select offerings: service + SLA agreement + pricing

Step Number	1.1.1
Step Description	SP service description + SLA specification
Precondition	SLA has been described for standalone VNF by the FPs
Involved T-NOVA Components	• SP Dashboard • Business Service Catalogue • SLA management module

Parameters	- SLA template
Test Methodology	The SP will perform the service description procedure involving the SLA template fulfilment by means of the connection to the SLA management module for different kind for services.
Metrics	- Time between the SP opening the service description GUI and the SLA template is available to be completed. - Time between the service description is completed by the SP and the notification that the service information is available in the Business Service catalogue and SLA module.
Expected Results	SLA template fulfilled by the SP and store in the SLA management module in a reasonable time.

Step Number	1.1.2
Step Description	The Customer browses the business service catalogue
Precondition	Service description and SLA specification for several services
Involved T-NOVA Components	- Customer dashboard - Business Service Catalogue - SLA management module
Parameters	- Search time
Test Methodology	The customer will introduce different search parameters
Metrics	Time since the customer introduces a search parameter until the system shows service options
Expected Results	The dashboard will have to show in a reasonable time the offerings available in the business service catalogue marching the parameters introduced

Step Number	1.1.3
Step Description	Customer selects an offering and accepts the SLA conditions
Precondition	The customer has performed search in the Business Service Catalogue
Involved T-NOVA Components	- Customer dashboard - Business Service Catalogue - SLA management module
Parameters	- Service selection
Test	The customer selects an offering what will imply that customer

Methodology	will have to accept several conditions coming from the SLA specification in the SLA module
Metrics	• Time since the customer selects and offering till the SLA conditions are shown to be accepted.
Expected Results	Conditions showed to the customer to be accepted in a reasonable time.

Step Number	1.1.4
Step Description	The SLA agreement is created and stored
Precondition	The customer has accepted the conditions associated to a given SLA specification
Involved T-NOVA Components	• Customer dashboard • SLA management module • Accounting
Parameters	• SLA agreement
Test Methodology	The SLA contract is signed by SP and customer and store in the SLA module. (the price will be store in the accounting).
Metrics	• Time since the customer has accepted the applicable conditions till the SLA contract is store in the SLA module (including SLA parameters that will need to be monitored by the orchestrator monitoring system)
Expected Results	SLA agreement between customer and SP in a reasonable time

5.1.3. UC1.2 – Advertise NFs

Step Number	1.2.1
Step Description	FP uploads the packaged VNF, providing also the metadata information.
Precondition	The FP developer authenticates through the dashboard
Involved T-NOVA Components	Dashboard, Marketplace, NF Store
Test Methodology	Multiple uploads of VNFs (package and metadata) will be executed. Measure various metrics (below)
Metrics	Upload time, system response, NF Store specific database performance metrics.
Expected Results	Fast response of the dashboard for the uploading of the VNF Quick update of the service catalogues

Step Number	1.2.2
Step Description	FP monitors the VNF use, popularity, reputation
Precondition	VNF is advertised and used by Customers
Involved T-NOVA Components	Dashboard, Marketplace, NF Store
Test Methodology	Test the responsiveness and the error free operation of the monitoring component of the dashboard system
Metrics	• Usability • Responsiveness • Update rate
Expected Results	The dashboard updates the monitoring information frequently according to FP selections, the interface is usable and responsive.

5.1.4. UC1.3 – Bid / trade

Step Number	1.3.1
Step Description	SP trades via brokerage platform with FPs
Precondition	The Customer has selected a NS that is offered via Service Catalogue and requires brokerage.
Involved T-NOVA Components	• Customer dashboard • SLA management module • Brokerage module
Parameters	Function Price, SLA agreement, Service Description.
Test Methodology	Validate that the returned NS is always the best fit to the Customer requirements.
Metrics	• Time since the customer sends the requirement till the system returns the NS • Profit margin for the SP as result of the transaction
Expected Results	Brokerage platform returns the appropriate NS matching the requirements set by the Customer.

5.1.5. UC2 – Provision NFV services / Map and deploy services

Step Number	2.1
Step Description	Provision NFV service
Precondition	Through the customer portal, the T-NOVA Customer has selected service components and relevant parameters (UC 1)

Involved T-NOVA Components	IVM, orchestrator, VNF
Test Methodology	Measure time between service request and the moment the service is fully operational (how to verify that the service is operational depends on the specific VNF)
Metrics	Metrics to verify: time to setup and activate the service from the moment the request is submitted by the customer, data plane performance (e.g. throughput, e2e delay), control plane performance (VNF-specific), time taken to enforce changes submitted by the customer
Expected Results	Success criteria - the service is fully operational after the NFV service provisioning sequence.

5.1.6. UC3 – Reconfigure/Rescale NFV services

Step Number	3.1
Step Description	A scale of a VNF will need to change in accordance with the traffic load profile. Traffic threshold defined in the SLA associated with the VNF will define the network traffic levels.
Precondition	Service monitoring provides metric data on a VNF to the SLA Monitor component The SLA Monitor detects that the SLA associated with the VNF is approaching a trigger threshold. The SLA Monitor determines the require action based on the associated SLA. The SLA Monitor notifies the Reconfigure/Rescale Service NFV of the scaling action required
Involved T-NOVA Components	• Virtual Infrastructure Manager (VIM) • NFVI • Orchestrator
Parameters	VNF specific. Likely parameter will be: • Network traffic load • Number of concurrent users • Number of concurrent sessions • Throughput or latency
Test Methodology	1. Select SLA parameter and specify a threshold, which can be, breached e.g. network traffic load in the first VNF of the service chain. 2. Use network traffic generator to generate load below SLA threshold level. 3. Increased the traffic load in step wise manner up to the

	threshold. 4. Monitor VIM to determine if new VM is added to OpenStack environment and to the correct VLAN. 5. Monitoring network traffic latency/throughput has been reduced.
Metrics	• Accuracy and validity of rescale decision • Time delay from the variation of the metric until the rescale decision • Time delay from the rescale decision to the completion of the rescaling of the service • Service downtime during rescaling
Expected Results	• VNF is scaled as per SLA threshold conditions • Additional VNF VM functions correctly as measured by the expected impact on the trigger boundary condition e.g. network latency/throughput. • Service downtime stays at minimum

5.1.7. UC4 – Monitor NFV services

Step Number	4.1
Step Description	Monitor NFV Service - i) Measurement process
Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded
Involved T-NOVA Components	• VNF • NFVI • VIM
Test Methodology	1. Feed a node port with artificially generated traffic with known parameters, 2. Artificially stress a VNF container (VM), consuming its resources by a mock-up resource-demanding process
Metrics	Observe measurements collected by the VIM Monitoring Manager. Performance indicators to be observed: • accuracy of measurement • response time
Expected Results	• Metrics are properly propagated and correspond to the known traffic parameters and/or stress process • Response time is kept down to the minimum

Step Number	4.2
Step Description	Monitor NFV Service - ii) Communication of metrics to Orchestrator

Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded, Step 4.1 has been completed
Involved T-NOVA Components	• VNF • NFVI • VIM • Orchestrator
Test Methodology	1. Feed a node port with artificially generated traffic with known parameters 2. Artificially stress a VNF container (VM), consuming its resources by a mock-up resource-demanding process
Metrics	Observe response time i.e. time interval from the change in resource usage until the Orchestrator becomes aware of the change
Expected Results	• Metrics are properly propagated and correspond to the known traffic parameters and/or stress process • Response time is kept down to the minimum

Step Number	4.3
Step Description	Monitor NFV Service - iii) Communication of alarms to Orchestrator
Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded, Step 4.2 has been completed.
Involved T-NOVA Components	• VNF • NFVI • VIM • Orchestrator
Test Methodology	1. Manually fail a network link 2. Manually drain VNF container resources 3) Artificially disrupt VNF operation
Metrics	Observe the updates in the Orchestrator monitoring repositories and measure accuracy and response time i.e. time interval from the change in resource usage until the Orchestrator records the change
Expected Results	• Metrics are properly propagated and correspond to the known traffic parameters and/or stress process • Response time is kept down to the minimum

5.1.8. UC4.1 - Monitor SLA

Step Number	4.4
Step Description	Monitor SLA

Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded,
Involved T-NOVA Components	• Orchestrator • Marketplace
Test Methodology	Follow procedure similar to UC4.2 (artificially consume and drain VNF resources) and/or UC4.3 (disrupt service operation). Validate that SLA status is affected.
Metrics	Measure SLA monitoring accuracy, especially SLA violation alarms. Measure response time (from the incident to the display of the updated SLA status on the Dashboard)
Expected Results	• Proper SLA status update • Proper indication of SLA violation • Minimum response time

5.1.9. UC5 – Bill NFV services

Step Number	5.1
Step Description	Billing for the service provider (SP) - i) NF has been registered and deployed
Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded
Involved T-NOVA Components	• VNF • NFVI • VIM • Marketplace
Test Methodology	1. Use the Marketplace to request deployment and provisioning of the NF
Metrics	-
Expected Results	NF has been successfully deployed (as reported in the marketplace dashboard)

Step Number	5.2
Step Description	Billing for the service provider (SP) - 1) NF usage data can be monitored
Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded
Involved T-NOVA Components	• Monitoring @ Marketplace level • Accounting • Marketplace
Test Methodology	1. Use the marketplace dashboard to check the resource usage by the deployed NF

Metrics	
Expected Results	Resource consumed data shown in the marketplace dashboard (some time after deployment)

Step Number	5.3
Step Description	Billing for the service provider (SP) - 1) NF billable terms and SLA elements can be accessed
Precondition	Service has been properly registered in the Marketplace
Involved T-NOVA Components	• Marketplace
Test Methodology	1. Use the marketplace interface to extract the NF billable metrics and SLA terms
Metrics	
Expected Results	Retrieve the list of billable items and SLA terms from the Marketplace store

Step Number	5.4
Step Description	Billing for the service provider (SP) - 1) Get the pricing formula for the NF at this provider
Precondition	Service has been properly registered in the Marketplace
Involved T-NOVA Components	Marketplace
Test Methodology	1. Use the marketplace interface to extract the NF pricing / billing model
Metrics	
Expected Results	Receive the pricing equation for the NF for the provider where it is deployed

Step Number	5.5
Step Description	Billing for the service provider (SP) - 1) Generate the invoice for a time period
Precondition	Service has been deployed i.e. UCs 1, 2 and 3 have preceded
Involved T-NOVA Components	Marketplace, Accounting
Test Methodology	1. Use the accounting interface to get the usage data for the period in question

Metrics	
	The invoice from the provider to the user for the NF and for the desired period is generated and available from the dashboard

5.1.10. UC6 - Terminate NFV services

Step Number	6.1
Step Description	A T-NOVA Customer terminates a provisioned service, over the T-NOVA dashboard.
Precondition	There is an existing active service running (deployed), for the specific customer.
Involved T-NOVA Components	Marketplace, Orchestrator, Billing
Parameters	N/A
Test Methodology	Dispatch termination request and observe the service status.
Metrics	Metrics to be verify: 1. Response time, to teardown the service 2. Update of associated information (duration of service, billing info, SLA data).
Expected Results	The resources used by this service, will be released. Billing information must be sent. In customer's marketplace view, this service will be shown as stopped.

Step Number	6.2
Step Description	A T-NOVA SP terminates all active services, that he owns.
Precondition	There are several services running (deployed), for different Customers.
Involved T-NOVA Components	Marketplace, Orchestrator, Billing
Parameters	N/A
Test Methodology	Measure time between discard action made and the moment that all services are fully deactivate. Measure the response time, in each component.
Metrics	Metrics to be verify:

	1. Response time, to discard the service, and inform involved actors (SP, Customers). 2. All needed information is stored correctly (duration of service, billing info, SLA data).
Expected Results	The resources used by the services, must be released. Billing information must be sent. In customer's marketplace view, this service will be shown as stopped. In SP's portal view, all services must be stopped.

Step Number	6.3
Step Description	A T-NOVA SP terminates a provisioned service, for a specific T-NOVA Customer.
Precondition	There is an existing active service running (deployed), for the specific customer.
Involved T-NOVA Components	Marketplace, Orchestrator, Billing
Parameters	N/A
Test Methodology	Measure time between discard action made and the moment the service is fully deactivate. Measure the response time, in each component.
Metrics	Metrics to be verify: 1. Response time, to discard the service, and inform involved actors (SP, Customer). 2. All needed information is stored correctly (duration of service, billing info, SLA data).
Expected Results	The resources used by this service, will be released. Billing information must be sent. In customer's marketplace view, this service will be shown as stopped. In SP's portal view, the service must be stopped..

5.2. Evaluation of T-NOVA VNFs

Apart from system-wide validation based on use cases, a separate evaluation campaign will be conducted in order to assess the efficiency and performance of the VNFs to be developed in T-NOVA, namely:

- Security Appliance (SA)
- Session Border Controller (SBC)
- Deep Packet Inspector (DPI)
- Home Gateway (HG)

The following figure presents a brief mapping of the VNFs to the taxonomy as provided by ETSI NFV ISG (see Section 2). This mapping assists the selection of the tools to be employed for the evaluation of each VNF.

Table 5 ETSI taxonomy mapping of T-NOVA VNFs

	Data Plane			Control Plane			Signal Processing	Storage	
VNF	Edge NF	Intermediate NF	Intermediate NF with Encryption	Routing	Authentication	Session Management	Signal processing	Non-Intensive	R/W Intensive
Security Appliance (SA)		X						X	
Traffic Classification (vDPI)		X		X					X
Session Boarder Gateway (vSBC)	X		X	X	X	X	X	X	
Home Gateway (vHG)	X			X	X	X		X	

The following chapters provide a description of some of the tools to be used for the validation of the specific VNFs.

5.2.1. Generic tools for validation and evaluation

5.2.1.1. Traffic Generators

(a) Non-Free

Enterprise level, non-free packet generators and traffic analyser software can be used in order to quickly and based on standard methodologies assess the system/component performance. However these tools are expensive and during the evaluation activities might not be available for use. For example, vendors such as Spirent and Ixia (mentioned in Section 2) already provide end-to-end testing solutions that deliver high performance with deterministic results. The solutions are based on hardware and software solutions capable of conducting repeatable test sequences utilizing a large number of concurrent flows containing a variety of L7.

(b) Open Source

Open Source community tools, are easier to access and compare results.

L2-L4 Traffic Generation tools

Pktgen

The pktgen software package for Linux [PKTGEN] is a popular tool in the networking community for generating traffic loads for network experiments. Pktgen is a high-speed packet generator, running in the Linux kernel very close to the hardware, thereby making it possible to generate packets with very little processing overhead. The packet generation can be controlled through a user interface with respect to packet size, IP and MAC addresses, port numbers, inter-packet delay, and so on. The pktgen is used to test network equipment for stress, throughput and stability behavior. Pktgen is included in the Linux kernel, thereby making it possible to generate packets with very little processing overhead. You can create a high-performance traffic generator/analyzer using Linux PC.

D-ITG

D-ITG (Distributed Internet Traffic Generator) [D-ITG] is a platform capable to produce IPv4 and IPv6 traffic by accurately replicating the workload of current Internet applications. At the same time D-ITG is also a network measurement tool able to measure the most common performance metrics (e.g. throughput, delay, jitter, packet loss) at packet level.

D-ITG can generate traffic following stochastic models for packet size (PS) and inter departure time (IDT) that mimics application-level protocol behavior. By specifying the distributions of IDT and PS random variables, it is possible to choose different renewal processes for packet generation: by using characterization and modeling results from literature, D-ITG is able to replicate statistical properties of traffic of different well-known applications (e.g Telnet, VoIP - G.711, G.723, G.729, Voice Activity Detection, Compressed RTP - DNS, network games).

At the transport layer, D-ITG currently supports TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP1 (Stream Control Transmission Protocol), and DCCP1 (Datagram Congestion Control Protocol). It also supports ICMP (Internet Control Message Protocol). Among the several features described below, FTP-like passive mode is also supported to conduct experiments in presence of NATs, and it is possible to set the TOS (DS) and TTL IP header fields. The user simply chooses one of the supported proto- cols and the distribution of both IDT and PS will be automatically set.

Pktgen-DPDK

Pktgen-DPDK [PKTGEN-DPDK] is a traffic generator powered by Intel's DPDK at 10Gbit wire rate traffic with 64 byte frames.

PFRING

PF_RING is a high-speed packet capture library that turns a commodity PC into an efficient and cheap network measurement box suitable for both packet and active traffic analysis and manipulation.

NETMAP

netmap / VALE is a framework for high speed packet I/O. Implemented as a kernel module for FreeBSD and Linux, it supports access to network cards (NICs), host stack, virtual ports (the "VALE" switch), and "netmap pipes". netmap can easily do line rate on 10G NICs (14.88 Mpps), moves over 20 Mpps on VALE ports, and over 100 Mpps on netmap pipes. netmap/VALE can be used to build extremely fast traffic generators, monitors, software switches, network middleboxes, interconnect virtual machines or processes, do performance testing of high speed networking apps without the need for expensive hardware. We have full support for libpcap so most pcap clients can use it with no modifications. netmap, VALE and netmap pipes are implemented as a single, non intrusive kernel module. Native netmap support is available for several NICs through slightly modified drivers; for all other NICs, we provide an emulated mode on top of standard drivers. netmap/VALE are part of standard FreeBSD distributions, and available in source format for Linux too.

MGEN

The Multi-Generator (MGEN) [MGEN] is open source software by the Naval_Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) group that provides the ability to perform IP network performance tests and measurements using UDP and TCP IP traffic. The toolset generates real-time traffic patterns so that the network can be loaded in a variety of ways. The generated traffic can also be received and logged for analyses. Script files are used to drive the generated loading patterns over the course of time. These script files can be used to emulate the traffic patterns of unicast and/or multicast UDP and TCP IP applications. The tool set can be scripted to dynamically join and leave IP multicast groups. MGEN log data can be used to calculate performance statistics on throughput, packet loss rates, communication delay, and more. MGEN currently runs on various Unix-based (including MacOS X) and WIN32 platforms. The principal tool is the mgen program, which can generate, receive, and log test traffic. This document provides information on mgen usage, message payload, and script and log file formats. Additional tools are available to facilitate automated script file creation and log file analyses.

IPERF

IPERF [IPERF] is a commonly used network-testing tool that can create Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) data streams and measure the throughput of a network that is carrying them. IPERF is a tool for network performance measurement and specifically for active measurements of the maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to timing, protocols, and buffers. For each test it reports the bandwidth, delay jitter, datagram loss and other parameters. IPERF is written in C.

IPERF allows the user to set various parameters that can be used for testing a network, or alternatively for optimizing or tuning a network. IPERF has client/server functionality, and can measure the throughput between the two ends, either unidirectionally or bi-directionally. It is open-source software and runs on various platforms including Linux, Unix and Windows (either natively or inside Cygwin).

- UDP: When used for testing UDP capacity, IPERF allows the user to specify the datagram size and provides results for the datagram throughput and the packet loss.

- TCP: When used for testing TCP capacity, IPERF measures the throughput of the payload. Iperf uses 1024 × 1024 for megabytes and 1000 × 1000 for megabits.

Typical IPERF output contains a time-stamped report of the amount of data transferred and the throughput measured.

IPERF is significant as it is a cross-platform tool that can be run over any network and output standardized performance measurements. Thus it can be used for comparison of both wired and wireless networking equipment and technologies. Since it is also open source, the user can scrutinize the measurement methodology as well.

Ostinato

Ostinato [OSTINATO] is an open-source, cross-platform network packet and traffic generator and analyzer with a friendly GUI. It aims to be "Wireshark in Reverse" and thus become complementary to Wireshark. It features custom packet crafting with editing of any field for several protocols: Ethernet, 802.3, LLC SNAP, VLAN (with Q-in-Q), ARP, IPv4, IPv6, IP-in-IP a.k.a IP Tunneling, TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD, HTTP, SIP, RTSP, NNTP, etc. It can import and export PCAP capture files. Ostinato is useful for both functional and performance testing.

The following table summarizes some of the most widely used traffic generators for L2-L4 assessment.

Table 6 Summary of L2-L4 Traffic Generators

Traffic Generator	Operating System	Network Protocols	Transport Protocols	Measured Parameters
Pktgen	Linux	IPv4,v6	UDP	Throughput
D-ITG	Linux, Windows	IPv4, IPv6	UDP, TCP, DCCP, SCTP, ICMP	Throughput, packet loss, delay, jitter
Pktgen-DPDK	Linux	IPv4,v6	UDP	Generation only
PFRING	Linux	IPv4,v6	UDP, TCP	Generation only
NETMAP	Linux, FreeBSD	IPv4,v6	UDP, TCP	Generation only
MGEN	Linux, FreeBSD, NetBSD, Solaris, SunOS, SGI, DEC	IPv4	UDP, TCP	Throughput, packet loss, delay, jitter
Iperf	Linux, Windows, BSD	IPv4	UDP, TCP	Throughput, packet loss, delay, jitter
Ostinato	Linux	IPv4, IPv6, IP-in-IP (IP Tunneling)	Ethernet, 802.3, LLC SNAP, VLAN (with Q-in-Q), ARP, TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD,	Statistics Window shows real-time port receive/transmit statistics and rates

			HTTP, SIP, RTSP, NNTP	

L4-L7 Traffic Generation tools

- **SIPp** [SIPp]: which is a free Open Source test tool/traffic generator for the SIP protocol. It includes a few basic SipStone user agent scenarios (UAC and UAS) and establishes and releases multiple calls with the INVITE and BYE methods. It can also read custom XML scenario files describing from very simple to complex call flows. It features the dynamic display of statistics about running tests (call rate, round trip delay, and message statistics), periodic CSV statistics dumps, TCP and UDP over multiple sockets or multiplexed with retransmission management and dynamically adjustable call rates.
- **Seagull** [SEAGULL]: Seagull is a free, Open Source (GPL) multi-protocol traffic generator test tool. Primarily aimed at IMS (3GPP, TISPAN, CableLabs) protocols (and thus being the perfect complement to SIPp for IMS testing), Seagull is a powerful traffic generator for functional, load, endurance, stress and performance/benchmark tests for almost any kind of protocol. In addition, its openness allows to add the support of a brand new protocol in less than 2 hours - with no programming knowledge. For that, Seagull comes with several protocol families embedded in the source code: Binary/TLV (Diameter, Radius and many 3GPP and IETF protocols), External library (TCAP, SCTP), and Text (XCAP, HTTP, H248 ASCII).
- **TCPReplay** [TCPREP]**:** is a suite of GPLv3 licensed utilities for UNIX (and Win32 under Cygwin) operating systems for editing and replaying network traffic which was previously captured by tools like tcpdump and Ethereal/Wireshark. It allows you to classify traffic as client or server, rewrite Layer 2, 3 and 4 packets and finally replay the traffic back onto the network and through other devices such as switches, routers, firewalls, NIDS and IPS's. Tcpreplay supports both single and dual NIC modes for testing both sniffing and in-line devices.

5.2.1.2. SDN Controller evaluation tools

A measurement framework for the evaluation of OpenFlow switches and controllers has been developed in Oflops [OFLOPS]. OFLOPS is an open framework for openflow switch evaluation. The software suite consists of two modules OFLOPS and Cbench. OFLOPS (OpenFLow Operations Per Second) is a dummy controller used to stress and measure the control logic of OpenFlow switches. On the other hand, Cbench emulates a collection of substrate switches by generating large numbers of packet-in messages and evaluating the rates of the corresponding flow-modification messages generated by the controller. As the source code of the framework is distributed under an open license it can be adapted to evaluate the performance of within the T-NOVA project.

5.2.1.3. Service/Resource mapping evaluation tools

AutoEmbed [DIETRICH13] was originally developed for the evaluation of various aspects of multi-provider VN embedding, such as the efficiency and scalability of embedding algorithms, the impact of different levels of information disclosure on VN embedding efficiency, and the suitability of VN request descriptions. The AutoEmbed framework supports different business roles and stores topology and request information as well as the network state in order to evaluate mapping efficiency. AutoEmbed includes an extendable library which supports integration of additional embedding algorithms which can be compared against a reference embedding, e.g. by using linear program optimization to find optimal solutions for different objectives, or by using a different resource visibility level. Request and topology information are exchanged using XML schemata and thus simplifies intercommunication with existing components. The evaluation can either be done online by using the GUI, or by further processing of the meta-statistics (.csv files) computed by AutoEmbed library.

Alevin

ALgorithms for Embedding of VIrtual Networks (ALEVIN) is a framework to develop, compare, and analyze virtual network embedding algorithms [ALEVIN]. The focus in the development of ALEVIN has been on modularity and efficient handling of arbitrary parameters for resources and demands as well as on supporting the integration of new and existing algorithms and evaluation metrics. ALEVIN is fully modular regarding the addition of new parameters to the VNE model.

For platform independence, ALEVIN is written in Java. ALEVIN's GUI and multi-layer visualization component is based on MuLaViTo [MULATIVO] which enables us to visualize and handle the SN and an arbitrary number of VNs as directed graphs.

5.2.2. VNF Specific validation tools

5.2.2.1. Application Classifier (vDPI)

In T-NOVA the vDPI shares common properties with its hardware based counterpart. Activities in the frame of IETF Benchmarking Methodology WG, have proposed benchmarking methodologies for such devices i.e. [Hamilton07] (more specific to media aware type of classification). The goal of this document is to generate performance metrics in a lab environment that will closely relate to actual observed performance on production networks. The documents aim in examining performance and robustness across the following metrics:

- Throughput (min, max, average, standard deviation)
- Transaction rates (successful/failed)
- Application response times
- Number of concurrent flows supported
- Unidirectional packet latency

The above metrics are independent of the Device under Test (DUT) implementation. The DUT should be configured as when used in a real deployment or typical for the use case where the device is intended. The selected configuration should be available along with the benchmarking results. In order to increase and guarantee repeatability of the tests, the configuration scripts and all the information resulting to the testbed setup should be made available. A very important issue for the benchmarking of content-aware devices is the traffic profile that will be utilized during the experiments. Since the explicit purposes of these devices vary widely but they all inspect deeply in the packet payload in order to support their functionalities, the tests should utilize traffic flows that resample to the real application traffic. It is important for the testing procedure to define the following application flow specific characteristic:

- Data Exchanged by flow, bits
- Offered Percentage of total flows
- Transport protocol(s)
- Destination port(s)

Planned Benchmarking Tests

1. **Maximum application session establishment rate** - Traffic pattern generation should begin at 10% of the expected maximum through 110% of the expected maximum. The duration of each test should be at least 30 seconds. The following metrics should be observed
   - Maximum Application Flow rate – maximum rate at which the application is served
   - Application flow duration – min/max/avg application duration as defined by [RFC2647].
   - Application Efficiency – Is the % ratio of the Bytes transmitted minus retransmitted over transmitted bytes, as defined in RFC 6349.
   - Application flow latency – min/max/avg latency introduced by the DUT

2. **Application Throughput** – determine the forwarding throughput of the DUT. During this test the applications flow through DUT at 30% of maximum rate.
   - Maximum Throughput – maximum rate at which all application flows completed
   - Application flow duration – min.max/avg application duration
   - Application efficiency – as defined previously
   - Packet Loss
   - Application flow latency
   -

3. **Malformed traffic handling** – to determine the effects on performance and stability that malformed traffic may have on DUT. The DUT should be under malformed traffic at all protocol layers (fuzzed traffic).
4.
5.

5.2.2.2. Session Border Controller (vSBC)

The vSBC incorporates two separate functions within a single device: the Interconnection Border Control Function (IBCF) for the signalling procedures and the Border Gateway Function (BGF) focused on the user data plane. Signalling procedures are implemented using the Session Initiation Protocol (SIP), while the data or use plane usually adopts Real time Transport Protocol (RTP) for multimedia content delivery.

The metrics that will be adopted to characterize the virtual SBC performance necessarily refer to the sessions it can establish, and generally cover three main aspects:

- the maximum number of concurrent sessions that can be established by the SBC
- the maximum session rate (expressed as the number originated/terminated session per second)
- the quality of service perceived by the end-users during audio/video sessions.

The provided quality of service is usually verified by analyzing a set of parameters evaluated in each active session. The basic parameters are related to network jitter, packet loss and end-to-end delay [RFC3550]. However, also instrumental measurements of ad hoc objective parameters should be performed. In particular, objective assessment of speech and video quality should be achieved, using, for instance, the techniques described in rec. ITU-T P.862 (Perceptual Evaluation of Speech Quality, PESQ) for audio, or following the guidelines given in ITU-T J.247 (Objective perceptual multimedia video quality measurement in the presence of a full reference) for video.

The metrics above summarized are strictly correlated. In fact, it must be verified that the maximum number of concurrent sessions and the maximum session rate can be achieved simultaneously. Moreover, the quality of service must be continuously monitored under loading conditions, to verify that the end-user perception is not affected. To this end, ad hoc experiments must be designed, for instance by analysing a few sample sessions, maintained always active during loading tests.

Finally, overloading tests will also be designed. The maximum session rate will be exceeded of a quantity equal to 10%; the overload condition will be maintained for a given time interval, and the removed. After a specified settling time, the vSBC will converge again to the nominal performance.

5.2.2.3. Security Appliance (vSA)

For the validation of the vSA VNF, a broad set of intrusion/attack simulators exists. Depending on the type of attacks that will be tested, different tools that could be used are:

- **Low Orbit Ion Cannon (LOIC)**: This is an open source application that can be used for stress testing and denial-of-service attack generation. It is written in C# and is currently hosted on sourceforge (http://sourceforge.net/projects/loic/) and GitHub (https://github.com/NewEraCracker/LOIC/).

- **Internet Relay Chat (IRC) protocol**: In case of Distributed DoS attacks, the master can use the IRC protocol to send commands to the attacking machines equiped with LOIC. The IRC protocol (described in RFC 2812) enables the transfer of messages in the form of text between clients.
- **hping (http://www.hping.org/)**: hping is a command-line oriented TCP/IP packet assembler/analyzer. The interface is inspired to the ping(8) unix command, but hping isn't only able to send ICMP echo requests. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features including: firewall testing and Port scanning. Hping works on the following unix-like systems: Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MacOs X, Windows.
- **SendIP (http://www.earth.li/projectpurple/progs/sendip.html)**: SendIP is a command-line tool to send arbitrary IP packets. It has a large number of options to specify the content of every header of a RIP, RIPng, BGP, TCP, UDP, ICMP, or raw IPv4/IPv6 packet. It also allows any data to be added to the packet. Checksums can be calculated automatically, but if you wish to send out wrong checksums, that is supported too.
- **Internet Control Message Protocol (ICMP)**: this protocol can be used to report problems occurred during the delivery of IP datagrams within an IP network. It can be utilized for instance when a particular End System (ES) is not responding, when an IP network is not reachable, or when a node is overloaded.
- **Ping**: The "ping" application can be used to check whether an end-to-end Internet Path is operational. Ping operates by sending Internet Control Message Protocol (ICMP) echo request packets to the target host and waiting for an ICMP response. In the process, it measures the time from transmission to reception (Round Trip Time - RTT -) and records any packet loss. This application can be used to detect whether a service is under attack or not. As an example, if a service is running in a virtual machine, checking the performance of the virtual machine through the RTT variation might show whether the service is under attack or not.

5.2.2.4. Home Gateway (vHG)

The Virtual Home Box integrates various middleware and service layer modules. Part of the proposed functionalities are related to video streaming, and therefore, it can also be viewed as a media server for End-Users.

Validation methodology for service environment (such as server monitoring) can be applied for vHG, to evaluate its performance as an individual entity during the content delivery and transcoding steps.

For testing the video quality at the user side, some standardized approaches exist. They will be used as performance metrics for validating video encoding/decoding and QoS/QoE estimation tools.

The validation method for Video Streaming will be built upon previous work carried out by some partners in the Alicante Project.

Peak Signal-to-Noise Ratio (PSNR)

For performing evaluations of the vHG one can use the well-known PSNR metric which offers a numeric representation of the fidelity of a frame/video. This metric is an objective measurement which allows an automatic calculation of the fidelity and, thus, needs not to be observed. PSNR allows the evaluation of the video quality resulting from decisions of the adaptation chain at the user environment.

Video Quality Metric (VQM)

The NTIA Video Quality Metric (VQM) [ALICANTE D8.1] is a standardized method of objectively measuring video quality by making a comparison between the original and the distorted video sequences based only on a set of features extracted independently from each video. The method takes perceptual effects of various video impairments into account (e.g., blurring, jerky/unnatural motion, global noise, block distortion, colour distortion) and generates a single metric which predicts the overall quality of the video.

Subjective Quality Evaluations

As the user environment is dedicated to the perceived quality of the service by the user, there is the need to perform subjective quality evaluations to effectively detect the quality of a system [ITU-RBT50013]. In this case, one can use a vast number of different evaluation methods such as Double Stimulus Continuous Quality Scale [PINSON04]. The DSCQS provides means for comparing two sequences subjectively. This means, that the user evaluates once a reference version (i.e., a version not processed by the system under investigation) and once a processed version (i.e., a version processed by the system under investigation). The given rating gives a feedback how well the system under investigation performs and if there is the need to adjust parameters.

6. CONCLUSIONS

Deliverable D2.51 presented the initial plan for the validation/experimentation campaign of T-NOVA. The target of experimentation has been the entire integrated T-NOVA system as a whole, rather than individual components. Taking into account the challenges in NFV evaluation, a set of system- and service- level metrics were defined, as well as the experimentation procedures for the validation of each of the T-NOVA use cases. The testbeds already available at the partners' sites, as well as the pilots to be integrated, constitute an adequate foundation for the assessment and evaluation the T-NOVA solution, under various diverse setups and configurations.

The experimentation/validation plan laid out in the present document will be subject to continuous elaboration throughout the project, depending on the progress of implementation, on the evolution of the technical architecture and the possible adjustment of technical details. It may also be affected by the continuous evolutions in the technical frameworks and especially the open-source projects (such as Openstack, OpenDaylight etc.) which have been selected as basis for the T-NOVA subsystems. Deliverable D2.52 (Planning of Trials and Evaluation – Final), due month 21, is expected to reflect all these evolutions and to present the final plan for the validation of the T-NOVA system.

7. References

[ALEVIN] ALEVIN: VNREAL, “ALEVIN – ALgorithms for Embedding VIrtual Networks,” May 2011. [Online]. Available: http://alevin.sf.netReference

[ALICANTE D8.1] Alicante Project "D8.1: Use Cases and Validation Methodology" http://www.ict-alicante.eu/validation/download/work-package/alicante_d8.1_v1.1.pdf

[BMWG] https://datatracker.ietf.org/wg/bmwg/charter/

[DIETRICH13] David Dietrich, Amr Rizk, and Panagiotis Papadimitriou. 2013. AutoEmbed: automated multi-provider virtual network embedding. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (SIGCOMM '13). ACM, New York, NY, USA, 465-466. DOI=10.1145/2486001.2491690 http://doi.acm.org/10.1145/2486001.2491690

[D2.1] T-NOVA Project, D2.1 “System Use Cases and Requirements”, 15th June 2014, on-line: http://www.t-nova.eu/wp-content/uploads/2014/11/TNOVA_D2.1_Use_Cases_and_Requirements.pdf

[D-ITG] Distributed Internet Traffic Generator, on-line: http://traffic.comics.unina.it/software/ITG/

[NFVPERF] NFV ISG, NFV Performance & Portability Best Practices, ETSI, v1.1.1, June 2014, on-line: http://www.etsi.org/deliver/etsi_gs/NFV-PER/001_099/001/01.01.01_60/gs_nfv-per001v010101p.pdf

[Hamilton07] M. Hamilton, S. Banks, “Benchmarking Methodology for Content-Aware Network Devices”, draft-ietf-bmwg-ca-bench-meth-07.

[IPERF] IPERF, on-line: http://sourceforge.net/projects/iperf/

[IPPM] IETF IP Performance Metrics Working Group, on-line: http://datatracker.ietf.org/wg/ippm/charter/

[ITU-RBT50013] ITU-R Rec. BT.500-13, "Methodology for the subjective assessment of the quality of television pictures", 2012

[IXIA] IXIA Web Site, on-line: http://ixiacom.com

[IXIABRC] IXIA Breaking Point Score, On-line: http://www.ixiacom.com/sites/default/files/resources/datasheet/resiliency-score.pdf

[IXIAINF] IXIA Infrastructure Testing, on-line: http://www.ixiacom.com/solutions/infrastructure-testing/index.php

[MGEN] MGEN, on-line: http://cs.itd.nrl.navy.mil/work/mgen

[MULATIVO] M. Duelli, J. Ott, and T. Mu ̈ller, “MuLaViTo – Multi-Layer Visualization Tool,” Apr. 2011. [Online]. Available: http://mulavito.sf.net

[NETMAP] NETMAP, on-line: http://info.iet.unipi.it/~luigi/netmap/

[OFLOPS] Charalampos Rotsos, Nadi Sarrar, Steve Uhlig, Rob Sherwood, and Andrew W. Moore. 2012. OFLOPS: an open framework for openflow switch evaluation. In *Proceedings of the 13th international conference on Passive and Active Measurement* (PAM'12), Nina Taft and Fabio Ricciato (Eds.). Springer-Verlag, Berlin, Heidelberg, 85-95. DOI=10.1007/978-3-642-28537-0_9 http://dx.doi.org/10.1007/978-3-642-28537-0_9

[OSTINATO] OSTINATO, on-line: https://code.google.com/p/ostinato/

[PKTGEN] Packet Gen, on-line: http://www.linuxfoundation.org/collaborate/workgroups/networking/pktgen

[PKTGEN-DPDK] Packet Gen – DPDK, on-line: https://github.com/Pktgen/Pktgen-DPDK/

[PF-RING] PF-RING, on-line: http://www.ntop.org/products/pf_ring/

[PINSON04] M. H. Pinson and S. Wolf, “A new standardized method for objectively measuring video quality,” Broadcasting, IEEE Transactions on, vol. 50, no. 3, pp. 312–322, September 2004

[RFC1944] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 1944, May 1996, <http://www.rfc-editor.org/info/rfc1944>

[RFC2498] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2498, January 1999, <http://www.rfc-editor.org/info/rfc2498>.

[RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", RFC 2647, August 1999, <http://www.rfc-editor.org/info/rfc2647>.

[RFC2679] Villamizar, C., Alaettinoglu, C., Govindan, R., and D. Meyer, "Routing Policy System Replication", RFC 2769, February 2000, <http://www.rfc-editor.org/info/rfc2769>.

[RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999, <http://www.rfc-editor.org/info/rfc2680>.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999, <http://www.rfc-editor.org/info/rfc2681>.

[RFC2889] Mandeville, R. and J. Perser, "Benchmarking Methodology for LAN Switching Devices", RFC 2889, August 2000, <http://www.rfc-editor.org/info/rfc2889>.

[RFC3511] Hickman, B., Newman, D., Tadjudin, S., and T. Martin, "Benchmarking Methodology for Firewall Performance", RFC 3511, April 2003, <http://www.rfc-editor.org/info/rfc3511>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC6349] Constantine, B., Forget, G., Geib, R., and R. Schrage, "Framework for TCP Throughput Testing", RFC 6349, August 2011, <http://www.rfc-

editor.org/info/rfc6349>.

[SEAGULL] Seagull, on-line: http://gull.sourceforge.net/

[SIPp] SIPp, on-line: http://sipp.sourceforge.net/

[SPIRENT] Spirent Web Page, on-line: http://www.spirent.com

[TCPREP] TCP Replay, on-line: http://tcpreplay.appneta.com/

LIST OF ACRONYMS

Acronym	Explanation
AAA	Authentication, Authorisation, and Accounting
API	Application Programming Interface
CAPEX	Capital Expenditure
CIP	Cloud Infrastructure Provider
CSP	Communication Service Provider
DASH	Dynamic Adaptive Streaming over HTTP
DDNS	Dynamic DNS
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DoW	Description of Work
DPI	Deep Packet Inspection
DPDK	Data Plane Development Kit
DUT	Device Under Test
E2E	End-to-End
EU	End User
FP	Function Provider
GW	Gateway
HG	Home Gateway
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IP	Infrastructure Provider
ISG	Industry Specification Group
ISP	Internet Service Provider
IT	Information Technology
KPI	Key Performance Indicator
LAN	Local Area Network
MANO	MANagement and Orchestration

MVNO	Mobile Virtual Network Operator
NAT	Network Address Translation
NFaaS	Network Functions-as-a-Service
NFV	Network Functions Virtualisation
NFVI	Network Functions Virtualisation Infrastructure
NFVIaaS	Network Function Virtualisation Infrastructure as-a-Service
NIP	Network Infrastructure Provider
NS	Network Service
OPEX	Operational Expenditure
OSS / BSS	Operational Support System / Business Support System
PaaS	Platform-as-a-Service
PoC	Proof of Concept
QoS	Quality of Service
RTP	Real Time Protocol
SA	Security Appliance
SaaS	Software-as-a-Service
SBC	Session Border Controller
SDN	Software-Defined Networking
SDO	Standards Development Organisation
SI	Service Integrator
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SME	Small Medium Enterprise
SP	Service Provider
TEM	Telecommunication Equipment Manufacturers
TRL	Technology Readiness Level
TSON	Time Shared Optical Network
UC	Use Case
UML	Unified Modelling Language
vDPI	Virtual Deep Packet Inspection
vHG	Virtual Home Gateway
VM	Virtual Machine
VNF	Virtual Network Function

VNFaaS	Virtual Network Function as a Service
VNPaaS	Virtual Network Platform as a Service
vSA	Virtual Security Appliance
vSBC	Virtual Session Border Controller
WAN	Wide Area Network
WP	Work Package

List of Contributors