CELTIC Project Number:	**C2012/2-5**
Project Title:	SDN Concept in Generalized Mobile Network Architectures – SIGMONA
Confidentiality:	PU[1]

Document Identifier:	D2.3
Document Title:	**Validation report on network management, monitoring, and control architectures in SDMN**
Authors:	VTT
Participants:	6WIND, Avea, Coriant, EXFO, Montimage, Nokia Hungary, VTT
Work Package:	WP2
Version:	1.0
Date of last changes:	23.03.16
File Name:	D2.3 Validation report on network management, monitoring, and control architectures in SDMN.doc

Abstract: This document describes the validations of network management, control, and monitoring solutions proposed in WP2. The validation platforms, tools, methods, and results are presented.

Keywords: validation, testing, network management, network control, network monitoring, network virtualization, Software Defined Mobile Network, backhaul

[1] Dissemination level:
PU = Public
RE = Distribution to a group specified by the consortium
CO = Confidential, only allowed for members of the consortium

Table of Contents

Authors

Partner	Name	Phone / Fax / e-mail
6WIND	François-Frédéric Ozog	
	Phone:	
	e-mail:	ff.ozog@6wind.com
AVEA	Engin Zeydan	
	Phone:	+90 216 (987)6386
	e-mail:	engin.zeydan@avea.com.tr
Coriant	Juha-Petteri Nieminen	
	Phone:	+358 40 8673208
	e-mail:	juha-petteri.nieminen@coriant.com
EXFO	Jorma Ikäheimo	
	Phone:	+358 40 3010215
	e-mail:	jorma.ikaheimo@exfo.com
Montimage	Edgardo Montes de Oca	
	Phone:	+331 53 803577
	e-mail:	edgardo.montesdeoca@montimage.com
Nokia Hungary	Zoltán Vincze	
	Phone:	+36 20 977 7797
	e-mail:	zoltan.vincze@nokia.com
VTT	Tapio Suihko	
	Phone:	+358 40 5529646
	e-mail:	tapio.suihko@vtt.fi
	Jori Paananen	
	Phone:	+358 40 7065621
	e-mail:	jori.paananen@vtt.fi

Executive Summary

This document reports the validation results of the SDMN concepts and solutions proposed in WP2. The validations concern research topics in the areas of

- network control and management:
  - hierarchical management and control of SDN-enabled RAN and backhaul
  - mobile backhaul traffic management and network performance optimization with SDN
  - virtualized hybrid-SDN backhaul network in multi-operator environments
  - secure backhauling
- monitoring
  - adaptation of a control plane protocol analyzer to virtual environments
  - DPI monitor virtualization in SDNs

For each solution proposal, the related usage scenarios, research challenges, functional objectives, and performance targets are described; the validation platforms, systems, tools, and methods are specified; the results are presented and analysed. The validations are still ongoing for the part of "DPI monitor virtualization in SDNs" listed above. Thus, the final results have not been available at the time of writing this document.

The validations confirm the applicability of SDN and virtualization techniques in mobile backhaul control and network monitoring. In the validations, SDN has enabled integrated control of RAN and backhaul resources. Virtualization, in terms of both hardware abstraction and network slicing, has also allowed embedding legacy last-mile backhaul network segments as part of a SDN-controlled network infrastructure, with automated incremental network deployment and infrastructure sharing among MNOs. The concept of control and data plane separation also entails the possibility of replacing expensive dedicated hardware servers with COTS equipment, which in the validations has been successfully used in securing communications between small cells and network aggregation points.

The virtualization of very high-performance live network analyzers has turned out slightly problematic. When a "real" network analyzer was executed in a virtual machine, the data capture performance suffered substantially. Virtualization of network monitoring systems raises also questions about the most feasible location of the monitoring probes. The probes should be distributed in the virtual machines, but they still require a centralized coordinator for supervising the monitoring tasks.

List of abbreviations

AAA	Authentication, Authorization and Accounting
AP	Access Point
API	Application Programming Interface
BSS	Business Support System
CoMP	Coordinated Multipoint
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPI	Deep Packet Inspection
eICIC	enhanced Inter-Cell Interference Coordination
EMS	Element Management System
eNB	eNodeB
EPC	Evolved Packet Core
FW	Firewall
GUI	Graphical User Interface
GW	Gateway
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
ISAAR	Internet Service quality Assessment and Automatic Reaction
KPI	Key Performance Indicator
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LTE	Long Term Evolution
MME	Mobility Management Entity
MMT	Montimage's monitoring solution
MNO	Mobile Network Operator
MPLS	Multiprotocol Label Switching
NFV	Network Function Virtualization
NFVI	Network Function Virtualisation Infrastructure
NM	Network Monitoring
NS3	Network Simulator version 3
NV	Network Virtualization
NVGRE	Network Virtualization using Generic Routing Encapsulation
OF	OpenFlow
ONF	Open Networking Foundation
OSS	Operations Support System
OVX	OpenVirteX
PBB	Provider Backbone Bridges
PC	Personal Computer
PCRF	Policy and Charging Rules Function
PDN	Packet Data Network
PE	Provider Edge
PGW	PDN Gateway
QMON	QoE Monitoring
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
REST	Representational state transfer

SAE	System Architecture Evolution
SDMN	Software-Defined Mobile Network
SDN	Software-Defined Network(ing)
SGW	Serving Gateway
SLA	Service Level Agreement
TNO	Transport Network Operator
vAnalyzer	Virtual analyzer
VM	Virtual Machine
VNE	Virtualized Network Element
VNF	Virtualized Network Function
VS	Virtual Switch
WLAN	Wireless LAN
WMN	Wireless Mesh Network
XDPd	eXtensible OpenFlow DataPath daemon
XDR	External Data Representation

List of Figures

1. Introduction

This document describes the validation of the network management, control, and monitoring solutions proposed in WP2. The mentioned validation activities can be divided into two major topics: (1) network management and control and (2) network monitoring. The structure of the document follows this partitioning; section 2 introduces the first topic group and section 3 will introduce the second one. Finally, section 4 summarizes the validation results.

For each solution proposal, there is a

- description of the related usage scenarios, research challenges, functional objectives, and performance targets;
- specification of the validation platforms, systems, tools, and methods; and
- description of the validation results and an assessment of the applicability of the solutions and their contribution to WP2 goals and project work.

2. Network management and control solutions

This section describes the validation of network management and control solutions proposed in WP2.

2.1 Integrated RAN/SDN controller for load balancing mobile backhaul networks

This validation presents an integrated scheme for load balancing in Software-Defined backhaul serving as the bitpipe for wireless access aggregation. The RAN and SDN controllers operate to optimize wireless access performance and traffic load in the system according to the dynamic properties of wireless segment and backhaul.

2.1.1 Validation objectives

The main objective of this validation is to demonstrate the proof of concept of the proposed SDMN mechanisms based on the theoretical results studied within WP1 & WP2.

For the validation platform of AVEA, mobile backhaul, which is composed of OpenFlow version 1.3 [OF13] capable SDN switches and controllers with OpenDaylight SDN platforms [OD], is used. Moreover, AVEA's testbed platform, constructed using mobile radio access controller, called RAN controller, and mobile backhaul controller, named as backhaul SDN controller, is integrated with the common Turkish consortium's testbed.

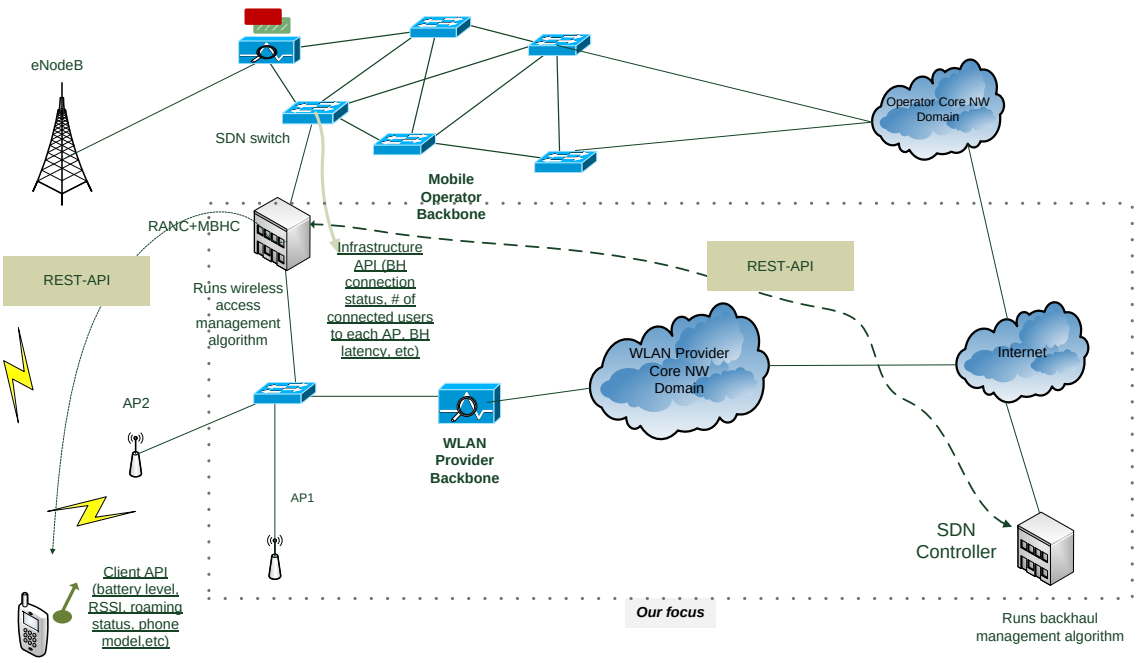

Figure 1: RAN Controller and Backhaul Management with hierarchical SDN-enabled mobile network infrastructure

The hierarchical system for managing and controlling the SDN-enabled mobile operator infrastructure with the perspective of load management is shown in Figure 1. The OpenDaylight based RAN and SDN controllers provide control functionality for managing backhaul and radio access resources (capacity, latency). The local RAN and SDN controllers dynamically re-configure the RAN and backhaul network. They reveal data and methods to control applications through an open northbound API. The main purpose of their actions is load management which is a critical requirement for scalable software-defined mobile networks.

OpenFlow and proprietary backhaul interface system APIs are used in the SDN controller to configure SDN enabled switch(es). The REST interface, which handles measurements and commands, is used as the northbound API and also for communication between the RAN and SDN controllers. The main functionality of the SDN controller is to manage load on different backhaul links to provide better quality of experience to the mobile users.

One of the key targets for the validation is to demonstrate the modularity of the proposed hierarchical SDN-enabled mobile operator architecture and control applications. We demonstrated the stand-alone operation of RAN controller and SDN controller during experiments in validation work. In other words, either the RAN controller or the SDN Controller can be used without deploying the other in the considered test environment.

2.1.1.1 Key success criteria and performance indicators

The key success criterion for our validation was to demonstrate the effectiveness and feasibility of the proposed load balancing approach. Moreover, load balancing related performance metrics were investigated to perform performance evaluation of the proposed scheme.

2.1.2 Validation environment

AVEA's and Turkish Consortium's joint validation platform consists of the following real network devices/elements:

- **Provided by AVEA:** Small cells (e.g. femto and Wi-Fi APs), OpenFlow capable Centec V350 switches [CEN] used as backbone switch sets, Cluster of Servers at R&D Test Lab (can be used for SGW and PGW, MME emulating servers, radio network controller server, network virtualization server and servers for operator domain IP services).
- **Provided by Turkish Consortium:** interconnected SDN infrastructure including different SDN controllers with various functionalities depending on use case.

2.1.3 Validation scenarios, methods and results

To examine the results and the performance of our approach, we implemented the system components, integrated them, set up a test system and ran experiments in this environment. OpenDaylight - Helium version was selected as the controller. HP 3800-48G-4XG switches were used in the experimental testbed. The wireless network segment was an IEEE 802.11 network with wireless access points and smart end-user devices including laptops and smart phones with IEEE 802.11 wireless interfaces.

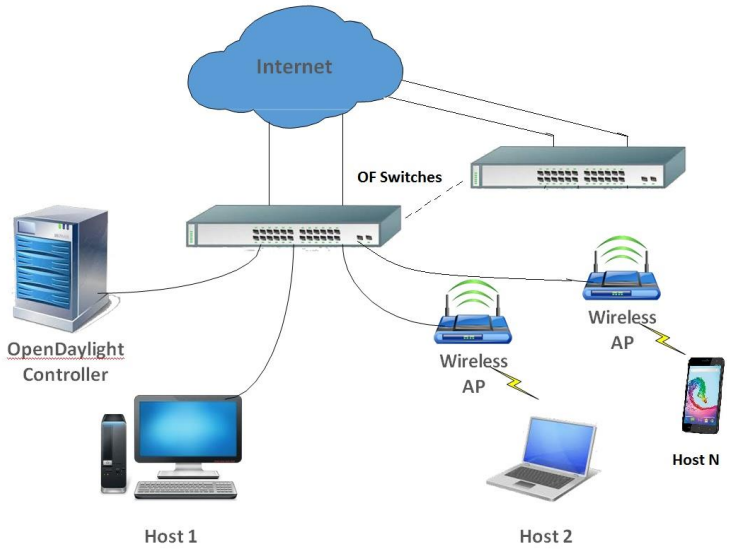

Figure 2 Experimental testbed for integrated load management scheme

The key objective, which was to demonstrate and investigate our proposed approach, was achieved with the experiments in our test environment. The validation work also led to the demonstration of our integrated load management concept in the envisaged software-defined mobile network.

2.1.4 Some Validation Results for Backhaul Segment

In this section, we provide some experimental results for our load management approach focusing on the backhaul segment in the SDMN environment. For this evaluation, we considered Abilene topology. It has hosts connected to switches that are connected to controllers which serve as SDN controllers for these nodes. There is a *super-controller* that is connected to the lower-tier controllers. This hierarchical structure works in a cooperative manner which entails information exchange for load metrics and partitioning. The hosts connected to switches generate flow requests according to a Poisson process where packet interarrival times are exponentially distributed. The packet sizes are assumed to be identical. The flows between switch pairs constitute flow- request information.

For reassignment and allocation, we need to keep the system performance high and the reassigning cost low. Therefore, we use a cost-based greedy algorithm after imbalance detection. The objective is to keep the total number of flow requests handled by any controller below a given threshold. After the switch pairs of average flow-request matrices are sorted in descending order of their load, the cost function is calculated for each pair as follows:

$$C_s = \overline{Q_s} + P_{ij}^s + \alpha v_M D_{is}$$

where Q_s is the average number of flow requests already handled by controller s at last two T_s periods, P^s_{ij} is the number of flow requests that is about to be handled by that controller and calculated by projecting B_{ij} on to the period of super controller's period, and D_{is} is the delay from switch i to controller s. The load balancing algorithm sorts the switch pair flows in descending order if imbalance is detected and reassigns the flows to the lowest-cost controller till the lower threshold is expected to be met.

We compare our algorithm with the random (RND) and the nearest neighbour (NEN) reassignment counterparts as baseline algorithms. Since any switch pair causing overload on any controller is reallocated to another randomly selected controller and to the physically closest controller without regard to its load and capacity in these algorithms, our algorithm is expected to perform better than they do. Our tests verify this expectation. In our experiments, we first set system and simulation parameters to the values shown in the table below.

Parameter	Value
number of nodes (N_n)	10
number of hosts (N_h)	10
number of controllers (N_c)	3
number of switches (N_s)	10
controller period (T_c)	10 msec
super controller period (T_s)	1000 msec
timeout period for flow entry in rule table	6 msec
host traffic generation rate (exponential (λ))	50
weight of past B_{ij} at t=-1 (w)	0.3
weight of past B_{ij} at t=-2 (z)	0.2
coefficient for cost function (α)	0.05
imbalance detection upper threshold (L_{upper})	1
imbalance detection lower threshold (L_{lower})	0.7

If RND algorithm is applied, the network stays in imbalanced state almost eight times more in comparison to when our algorithm is used. Moreover, the load distribution is altered without considering the individual capacities of controllers. Although the load figures in Figure 3.b may seem like a good balancing situation, the load on Controller 2 (C2) should stay higher since it has more capacity compared to C1 and C3. If the reassignment is applied with NEN algorithm, staying in imbalanced state decreases according to random algorithm but it is still almost six times more compared to our algorithm. Since there are three controllers in our topology and C1 is closer to C2, the system burden oscillates between these two controllers. However, because of its high capacity, C2 can serve more load than C1. Therefore, our algorithm performs better than RND and NEN algorithms.

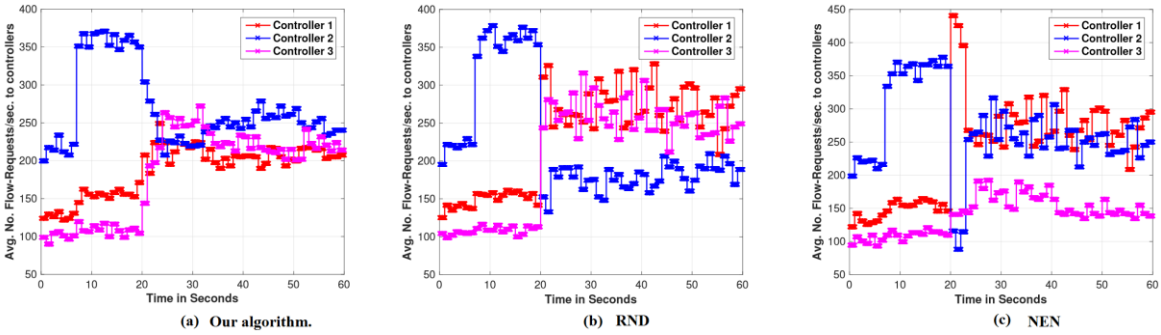

Figure 3: Experimental results for load management in backhaul

2.1.5 Summary

The validation objectives were to develop and demonstrate an integrated load management scheme considering wireless access network and mobile backhaul which was performed in an experimental testbed shown in Figure 1. The outcome of this work is also to be presented at IEEE/IFIP Network Operations and Management Symposium (IFIP NOMS 2016) as a demo paper.

2.2 Mobile backhaul management and optimization with SDN

The SON for MBH solution proposed in [SIG2] creates a self-configuring and self-optmizing mobile backhaul that is able dynamically and autonomously adjust itself to the actual traffic demand. The validation work introduced in the next subsection was focusing on showing that the SON for MBH solution is really able to achieve such network operation.

2.2.1 Validation objectives

The aim of the validation activity is to showcase that SDN backhaul provides means for rapid and consistent reconfiguration of backhaul services. It also proves that the flexible backhaul management offered by SDN creates the possibility of fast and automated optimization of LTE backhaul networks that can be used for maintaining adequate system QoS/QoE and rationalized resource usage at the same time.

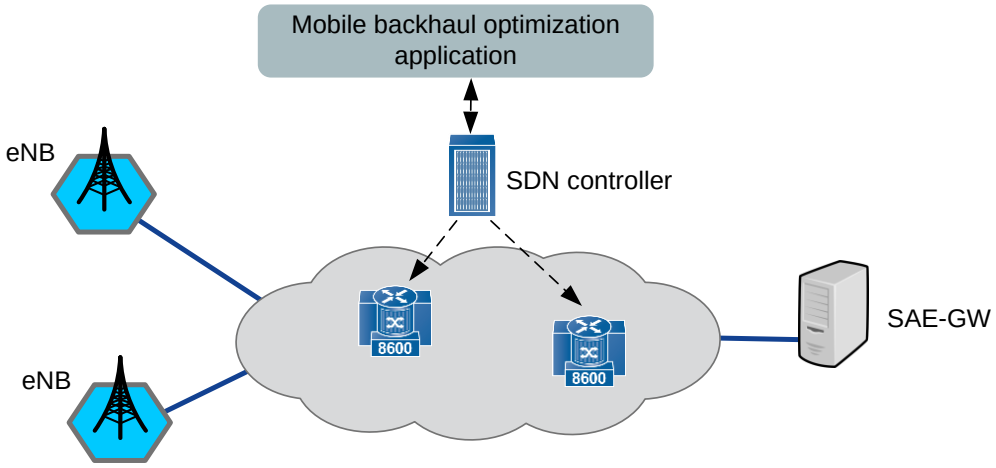

Figure 4: Mobile backhaul management and optimization with SDN

2.2.1.1 Key success criteria and performance indicators

The key success criterias are:

- The system should be able to detect QoE degradation and executue correction actions in short time
- The system should be able to create dedicated transport service for newly added eNB dynamically

2.2.2 Validation environment

A real LTE network with SDN backhaul network was created as the validation platform. The platform consisted of two commercial eNBs and one SAE-GW, Juniper site routers, Coriant SDN routers, and a Coriant SDN controller (see Figure 5). Traffic was generated using LTE USB dongles accessing public internet via the LTE network.

In the validation platform, the eNBs were connected to the SAE-GW via the mobile backhaul composed of SDN capable routers which were controlled by the SDN controller (see Figure 4). The SDN controller provides an HTTP/REST based API for managing (creating, updating, deleting) backhaul services and exposing topology information. This API is used by the SON for MBH solution to learn the network topology and the actual path of the services in the network. The SON for MBH conducts user- and control-plane measurements, QoE and QoS measurements; it continuously evaluates the status of the network and performs anomaly detection and localization, service or QoE degradation detection. If it detects that the backhaul needs to be optimized, it identifies the adequate steps and initiates the necessary backhaul management actions using the API of the SDN controller.

The validation platform consisted of the following real network devices/elements:

- Provided by Nokia: eNBs [FB], SAE-GW [FG], Mobile backhaul optimization application
- Provided by Coriant: 8600 MPLS routers [CO], SDN controller

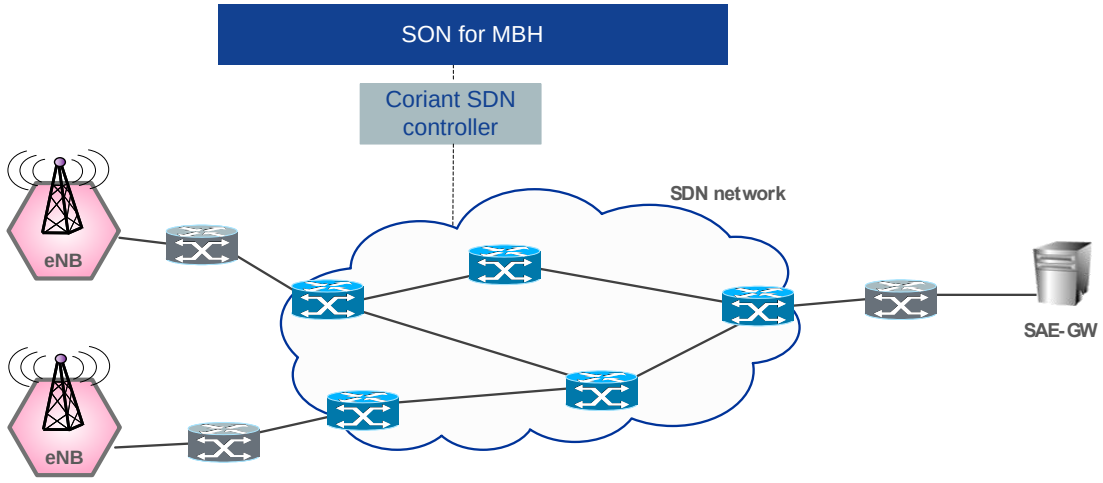

Figure 5: SON for MBH proof-of-concept setup

2.2.3 Validation scenarios, methods and results

The SON for Mobile Backhaul solution was validated in two main scenarios:

- Automated service creation for newly added eNBs
- Dynamic transport resource management following changing traffic demand

The next subsections describe the methods used in the scenarios and the results of the validation work.

2.2.3.1 Automated service creation scenario

The eNB auto-configuration SON use-case aims at delivering plug-and-play style deployment of new eNBs. Once the eNB is deployed and it starts up, it connects to the network management system via a basic transport service and downloads the proper radio and transport configuration. Once the configuration is applied, the eNB is ready for use and can connect to the SAE-GW via the required dedicated transport service. Such operation requires that the respective dedicated transport services are pre-planned and pre-configured.

The validation has shown that by using SON for MBH it is not needed anymore to pre-plan and pre-configure dedicated transport services for new eNBs; thus it makes it possible to create a true end-to-end plug-and-play process. In the performed validation scenario, one of the eNBs was decommissioned, reset to factory defaults. When the eNB was restarted, it started the auto-configuration process; this was detected by SON for MBH and it created the dedicated transport services using the SDN controller in parallel with the auto-configuration process. The SON for MBH was able to create the transport service on the path that was optimal under current network conditions. This is a big advantage compared to the case of pre-configured transport services as it is inevitable that the path determined by the planning

process based on traffic forecasts will not be globally optimal by the time the service is taken into use due to the prediction errors.

2.2.3.2 Dynamic transport resource management scenario

Using real-time measurement, analytics, decision and action, the SON for MBH is capable to dynamically optimize mobile backhaul. That results in that the network status is continuously adapted to the actual traffic demand and the resources are used utilized in optimal way.

To validate this capability, a scenario was setup where the transport service of the eNB was configured assuming low traffic demand. Once new users started to arrive to the eNB the transport service started to become overloaded and when the number of users was high enough to create congestion on the transport service, users started to experience QoE incidents. This undesired network state was detected by the SON for MBH within seconds and if the problem persisted for a while it took necessary corrective actions (e.g. increasing service bandwidth, rerouting service if there was no unused capacity left on the current path) using the SDN controller. This way, SON for MBH was able to dynamically redistribute the resources in the network based on the actual traffic demand. As Figure 6 shows, the benefits of such automated, dynamic network optimization is that the same transport infrastructure can accommodate more user traffic with the right quality, thus it can provide ~20% CAPEX gains.

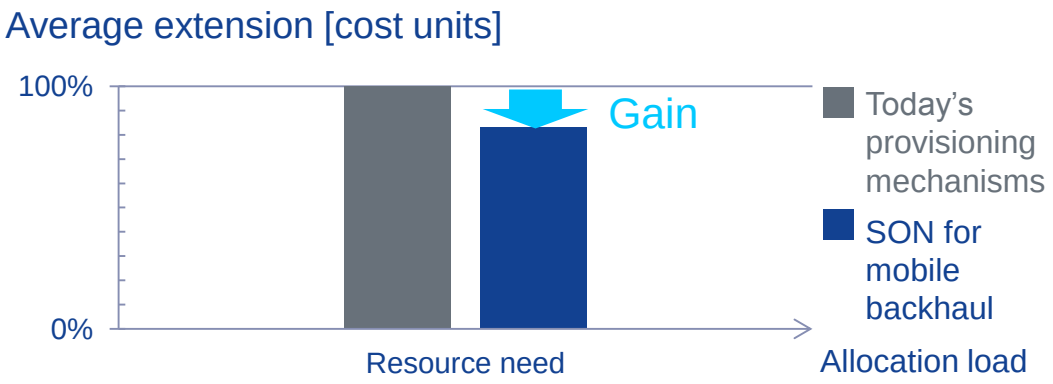

Figure 6: CAPEX gains provided by SON for MBH

2.2.4 Summary

The validation work reached its original goal, it could show that applying SDN to MBH networks combined with the SON for MBH solution (proposed in [SIG2]) makes it possible to build self-configuring and self-optimizing mobile backhaul networks. Using real-time measurement, analytics, decision and action, the validated system was able to detect and react to network anomalies. It also has been shown that such system supports end-to-end plug and play operation for newly added eNBs; that feature can significantly simplify network planning and operations.

2.3 Virtualized hybrid-SDN mobile backhaul

The purpose of the validation is to demonstrate and assess the feasibility of deploying a virtualized hybrid-SDN mobile backhaul network in multi-operator environments.

2.3.1 Validation objectives

The validation considers SDN-based mobile network backhaul infrastructure that would be provided by a transport network operator (TNO). The backhaul network resources may be shared among multiple mobile network operators (MNOs). Besides SDN switches, "legacy" network element technologies are deployed. The network's forwarding elements and the main centralised control plane functions are illustrated in Figure 7.

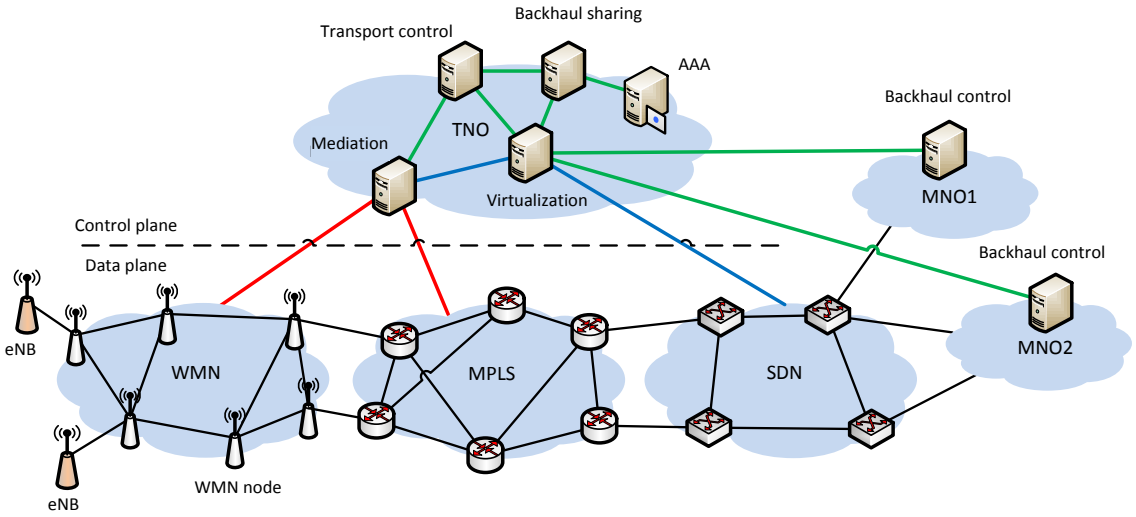

Figure 7: A schematic view to the virtualized hybrid-SDN mobile backhaul

The primary features and the main control functions in the system can be summarised as follows:

- TNO provides virtual network slices to authorised MNOs by using the "Backhaul sharing" management application
- MNOs (and the TNO itself) see their network slices through a FlowVisor ("Virtualization")
- MNO's slice grows incrementally and automatically as MNO installs new small-cell base stations into the network (e.g., pico or micro cells)
- "Mediator" provides a hardware abstraction layer that transforms legacy networks into one or more OpenFlow switches, i.e., the Mediator interprets OpenFlow Protocol and translates the operations to legacy control/management protocols.

The increased flexibility and rapidity in service creation and network resource sharing should allow reducing TNO's capital expenses and operational costs, through increased efficiency in network resource setup, usage and management. By using the network virtualization techniques, each MNO should be able to independently manage its own slice of the network, which allows fast reactivity to changing traffic demands.

The main challenges include support of plug-and-playability of small-cell base stations; development of viable SDN hardware abstractions of legacy network elements; construction of controller/virtualization hierarchy; and guaranteeing of carrier-grade forwarding performance, robustness and dependability of the control and data planes.

Besides data plane throughput, one of the key metrics is control plane latency (which sets a lower limit on the time it takes for a MNO to switch its traffic flow). The resource sharing functions and system dependability needs to be analysed.

2.3.1.1 Key success criteria and performance indicators

The main features of the system are automated base station commissioning and traffic management in a multitenancy environment, network survivability, co-existence with legacy transport equipment, and scalability:

- Base station deployment:
  - "zero-touch" base station and backhaul node installation
  - rapidity of base station commissioning
  - network attachment and resource access only to authorized and authenticated base stations
- Network control and management
  - traffic engineering
  - isolation across the network resource slices assigned to MNOs
  - manageability of the network slice assigned to MNO
- Network survivability:
  - no single points of failure
  - fast recovery from failures without packet loss
  - connection availability

- Co-existence with legacy transport equipment
  - interworking with legacy networks
- Scalability:
  - scalability wrt number of base stations
  - scalability wrt number of MNOs

2.3.2 Validation environment

The experimental evaluation was performed on a test network that models the network scenario shown in Figure 7. The test network consists of network domains of WMN and SDN technologies, computing platforms for the controller and management functions, and traffic generators. MPLS-switches were not included in the test setup.

The WMN is a self-organizing and self-optimizing transport network that adapts autonomously to changing traffic loads and link capacities. In the evaluation, it will be presented by VTT's existing WMN test bed, in which the millimetre wave radio links are emulated by using wired Ethernet links. The WMN nodes have been implemented by using Cavium Octeon network processors. For the SDN network emulation, *mininet* [MIN] virtual network was used.

The mediator was developed on the *Indigo Virtual Switch* [IND] framework, which itself uses the kernel module of *Open vSwitch* [OVS]. The *Ryu* [RYU] OpenFlow controller was used to provide the MNO's view to its network slice. *OVX* (i.e., *OpenVirteX*) [OVX] served for the SDN virtualization function. The backhaul control/management and AAA functions were running in Linux PCs.

Figure 8 gives a close-up view to the WMN and Mediator parts of the test setup for automated base station deployment.

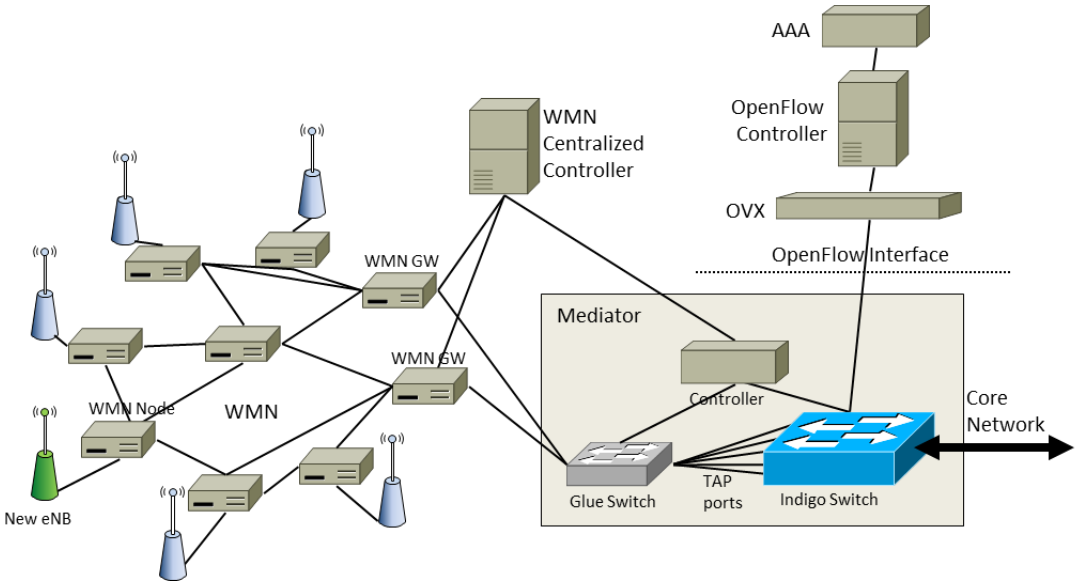

Figure 8: WMN and Mediator in base station deployment

The WMN gateway nodes are connected to the WMN Centralized Controller and the Mediator, which run in a Linux machine (or separate Linux machines). The base stations are emulated by laptops that connect to WMN nodes with Ethernet cables.

The Mediator implementation consists of three entities: The controller, the Indigo Virtual Switch modified to exchange information with the controller, and a front-end virtual switch (Glue Switch) confronting the WMN gateways. The Indigo Switch has an OpenFlow channel to an SDN controller via OVX hypervisor.

2.3.3 Validation scenarios, methods and results

The main validation scenario involves automated small-cell base station deployment (auto-enrolment), in which a small-cell base station is attached to one of the WMN backhaul nodes. The base station is automatically detected and it is allocated backhaul transport resources after verification of authentication and authorization by the AAA server. As a result, a port appears in the network slice of the MNO that owns the base station. Because the whole WMN is abstracted as a single virtualized switch, this means that a port appears in the switch in the MNO's network topology as seen from the MNO's OpenFlow controller. The flow of events in the auto-enrolment procedure is outlined in Figure 9.

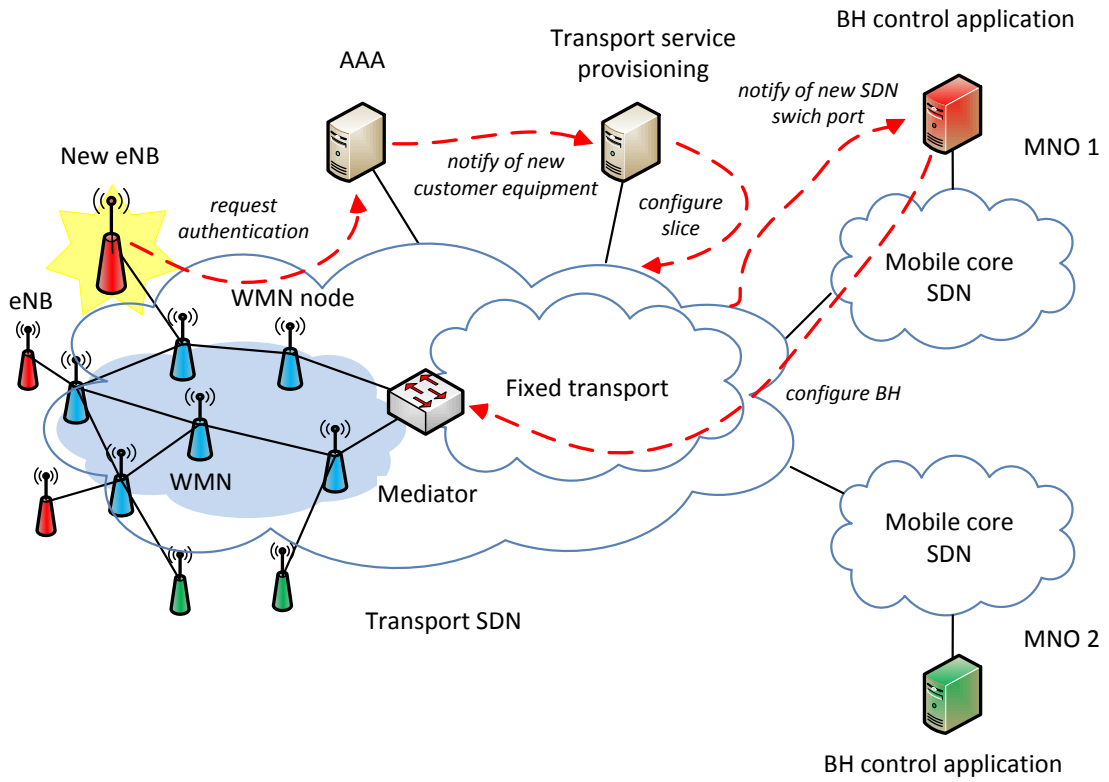

Figure 9: Small-cell base station deployment (auto-enrolment) scenario

The base station attached to a WMN node sends an authentication request as its first packet. The WMN node identifies it as an unknown packet (containing e.g. an unknown source MAC or VLAN id). The node has been configured to send unknown packets to one of the gateway nodes, possibly adding a new unique VLAN id to each packet, unless it already contains a VLAN id.

The authentication packet is received from the WMN gateway by the Mediator's Glue Switch. It detects the unknown VLAN id and creates a new TAP virtual network device for it. It commands the Indigo Switch to accept the device as a new port, removes the VLAN id from the packet and forwards the packet to the TAP device.

The Indigo Switch reports the creation of the new port and the reception of an unknown packet to the Mediator controller. It then sends the packet to the SDN controller in an OpenFlow Packet-In message, to be forwarded to the AAA server for authentication.

If the authentication succeeds, the controller sends two OpenFlow Add Flow messages to the Indigo Switch: one for outgoing (tap device as input port, an allocated port to the core network as output port) and one for incoming traffic (port roles reversed). The switch notifies the Mediator's controller of the added flows. The controller, in turn, informs the WMN Centralized Controller (WCC) of them. The WCC allocates a new virtual circuit in the WMN between the node where the base station was attached and a WMN gateway node. The WCC then sends the virtual circuit configuration to the nodes. The route between the base station and the core network is then established. Data plane operation efficiency on the path from the emulated base station and the core network has been verified by transporting a video stream along the path.

The solution scales well with respect to the number of base stations attached to the backhaul since each provisioning of a base station is reduced to the appearance of a port in the virtual switch. On the other hand, the abstraction of the whole WMN into a single virtual switch makes it difficult to provide enough isolation among MNOs' network slices. This is partly due to the simplicity of the OpenFlow switch model, but it is also an inherent property of the WMN, which is highly dynamic in its adaptation to capacity and traffic fluctuations without support for rigid resource reservations.

The resiliency of the WMN is supported by WMN-internal path redundancy and the availability of multiple WMN gateways. However, on the data plane, the presence of only one Mediator between the WMN and the fixed transport network causes a single point of failure. This issue cannot be solved just by adding a second Mediator in parallel, as it would undermine the WMN abstraction. One possibility for providing the required resilience is to exploit MC-LAG (Multi-Chassis Link Aggregation Group) functionality, that is available in legacy Ethernet switches, and hide that functionality inside the WMN abstraction. This is a viable option during the migration towards SDN since hybrid switches can be expected to support both SDN and traditional Ethernet mechanisms. Because SDN forwarding is not

based on Ethernet MAC address learning, as opposed to standard Ethernet forwarding, such application of link aggregation needs not be reliant on Link Aggregation Control Protocol (LACP), which greatly simplifies the solution.

2.3.4 Summary

Experimental evaluation with the virtualized hybrid-SDN mobile backhaul test network aimed at proving the feasibility of the SDN concept in a WMN-based backhaul, with less emphasis on measurable network performance metrics. The test setup focused on an OpenFlow-controlled last-mile WMN backhaul and its interworking with Ethernet technology.

The software programmability brings flexibility and rapidity in transport service creation. It also facilitates authenticated network attachment, slicing, and resource sharing among MNOs. Still, resource isolation across the MNOs' network slices is difficult to achieve when the WMN is abstracted to a single virtual switch (though, the isolation issue is inherent in the WMN as such).

On the other hand, SDN in its typical manifestations lacks the carrier-grade network qualities with respect to data plane survivability and control plane scalability. Data plane resiliency in the WMN abstraction can be achieved by exploiting link aggregation features that can be assumed available in hybrid Ethernet/SDN switches.

2.4 Secure backhauling

The purpose of this validation was to ensure cost effectiveness of securing communications between small cells and aggregation points near PE router.

2.4.1 Validation objectives

The main objective was to validate that COTS servers have enough encryption and routing capacity to sustain surging mobile bandwidth.

2.4.1.1 Key success criteria and performance indicators

- Encryption bandwidth should be above 100Gbps
- Minimum tunnel setup rate should be over 1000 per second

2.4.2 Validation environment

The benchmark tests measured performance using HTTP traffic over IPSec. The intent was to determine the maximum stable performance at reasonable CPU resource levels. The HTTP traffic model is designed by test vendor Spirent to imitate a real world scenario and provides a more reliable indicator of actual performance than simple "bit-blaster" performance tests.

Two server devices were placed under test: an HPE Proliant Gen8 with 60 cores, used mainly for decryption, and a Gen9 with 72 cores, used mainly for encryption. Both servers were running 6WIND Turbo IPSec software in VMs and 6WIND Virtual Accelerator software in the hypervisor (Figure 10).

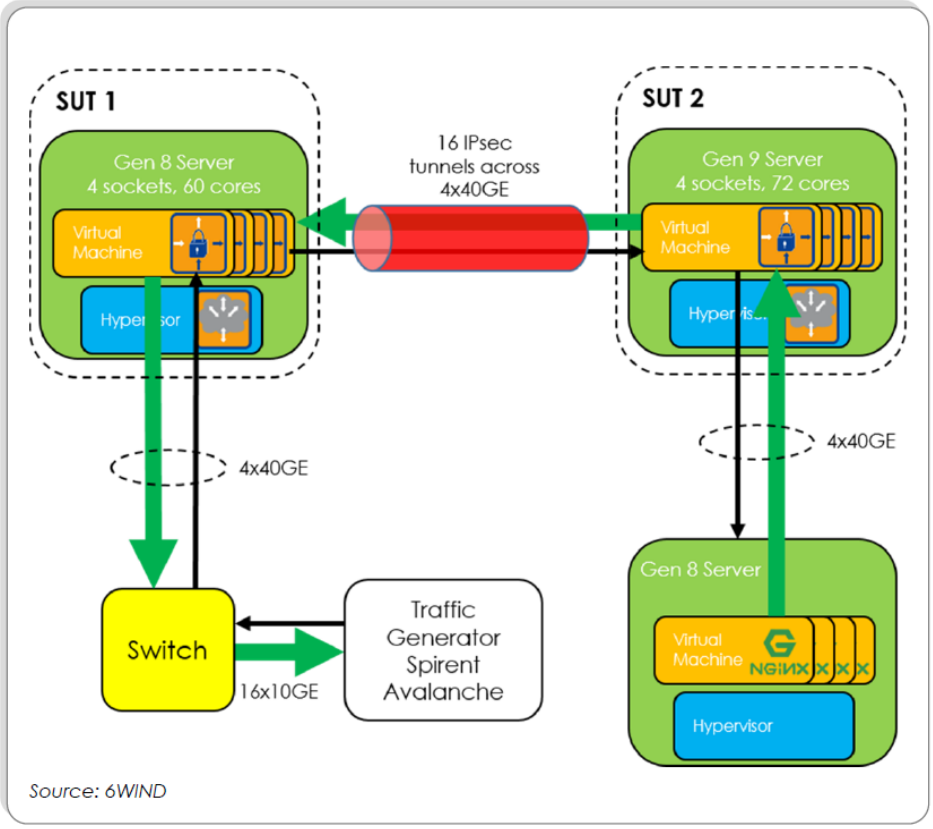

Figure 10: Logical tesbed setup

2.4.3 Validation scenarios, methods and results

In the test, the system showed 144Gbps of application traffic (AES-256) sustained at 75% percent of CPU utilisation (Figure 11).

System Components	HPE ProLiant DL580 Gen8 Server	HPE ProLiant DL580 Gen9 Server	Notes
Processor	Intel Xeon CPU E7-4890 v2 @ 2.80 GHz	Intel Xeon CPU E7-8890 v3 @ 2.50 GHz	4 sockets per system
Cores per system	60	72	
VM information	Turbo IPsec	Turbo IPsec	1 VM per socket, 4 VMs per system
Total cores for all (4) VMs	34	34	32 fastpath; 2 control
Total Virtual Accelerator cores	24	24	Part of hypervisor
Unused cores per system	2	14	Available for more VMs, performance
IPsec HTTP total bandwidth	144 Gbit/s	144 Gbit/s	IPsec uses AES-256 HMAC-MD5
CPU utilization	Turbo IPsec: 90% Virtual Accelerator: 70%	Turbo IPsec VM: 75% Virtual Accelerator: 55%	Nginx @ 100% utilization
IPsec HTTP BW/VM	144/32 = 4.5 Gbit/s	4.5 Gbit/s	32 cores in fastpath

Source: 6WIND

Figure 11: Performance test results

Note that the upper limit on performance was determined by Nginx Web server, running at 100% utilisation. The Gen9 server in particular could otherwise have achieved greater performance based on the CPU utilization rate. The full test used all four server sockets to achieve maximum performance, but the same tests were also run on a single, double and triple sockets, and demonstrated linear scalability from low to high performance.

2.4.4 Summary

This test not only proved that COTS servers have the capacity to aggregate large amounts of secure traffic but also compete with the high end dedicated hardware. Actually, the test bed matched the performance of the largest special purpose hardware on the market.

In addition, a cost study was conducted and showed that the cost per encrypted IPSec Gbps of COTS based solutions can be a fifth of the cost of dedicate hardware.

3. Network monitoring solutions

This section describes the validation of network monitoring solutions proposed in WP2.

3.1 Virtual analyzer

Virtual analyzer is based on EXFO's Power Hawk Pro probe software. It integrates to a virtual switch to capture packets for analysis. Virtual analyzer performs control plane call and session analysis and provides results out-of-band for various functions to perform network monitoring or troubleshooting tasks.

3.1.1 Validation objectives

The aim was to validate control plane protocol analyzer (vAnalyzer) performance and functionality when the analyzer is installed and executed in a virtual machine in OpenStack cloud environment (as shown in Figure 12) mimicking NFV environment (as shown in Figure 13).

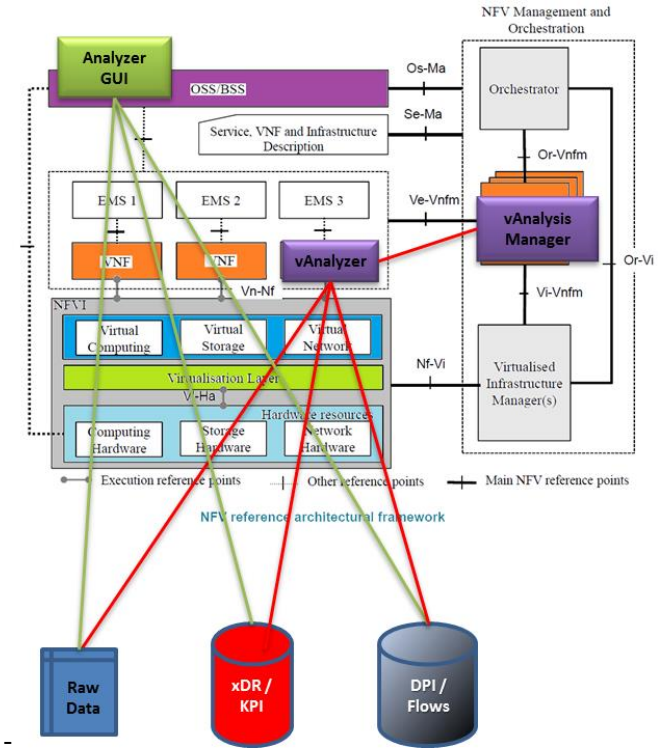

Figure 12: Virtual analyzer deployment to OpenStack host

The output of the validation involves a performance comparison between the virtual analyzer and a legacy analyzer.

3.1.1.1 Key success criteria and performance indicators

1. Capacity of data capture from physical Ethernet line
2. UDP and TCP bandwidth between virtual machines in an OpenStack server
3. Scalability in terms of packet trace analysis capacity per number of CPU cores

3.1.2 Validation environment

Validation was performed in OpenStack cloud environment. A traffic generator was used to send emulated network traffic to the system under validation. In the performance analysis, traffic rate in bits per second, sessions per seconds, CPU load, and memory usage were monitored and compared to those in a legacy analyzer. Mirror ports were used to capture the traffic to the respective analyzer ports because OpenStack environment does not support virtual tap ports for capturing traffic between virtual machines or from physical network ports.

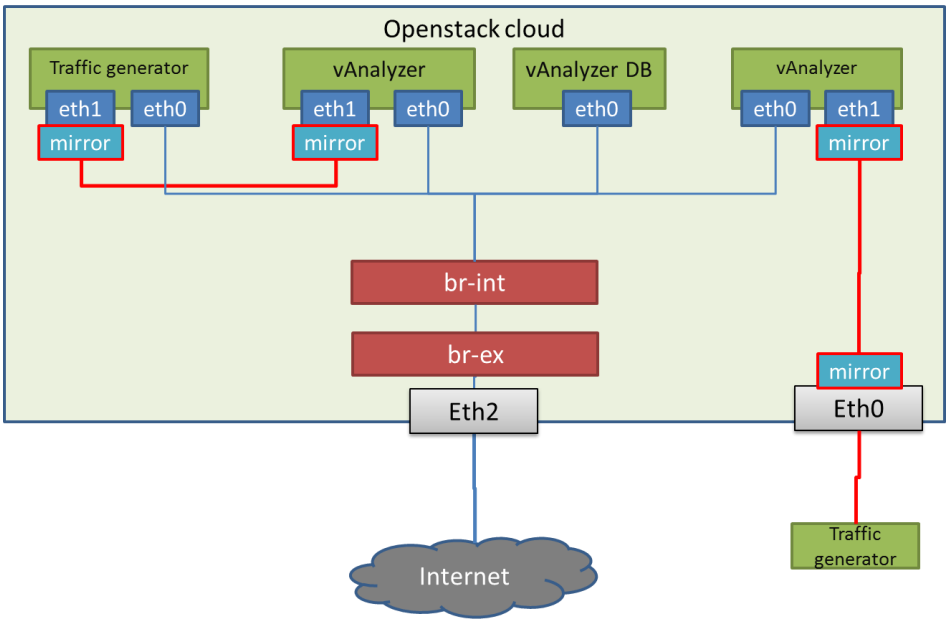

Figure 13: Validation environment

3.1.3 Validation scenarios, methods and results

3.1.3.1 Capture from physical Ethernet line

In this test case, we examined the ability to capture traffic from a physical Ethernet line to memory in different interfaces used in the system. The host system was equipped with Intel Xeon 5540 processor at 2,53 GHz, 1Gbit Ethernet adapter and it was running Ubuntu Linux, KVM, and OpenStack Juno release. The results of the tests are shown in Figure 14 below.

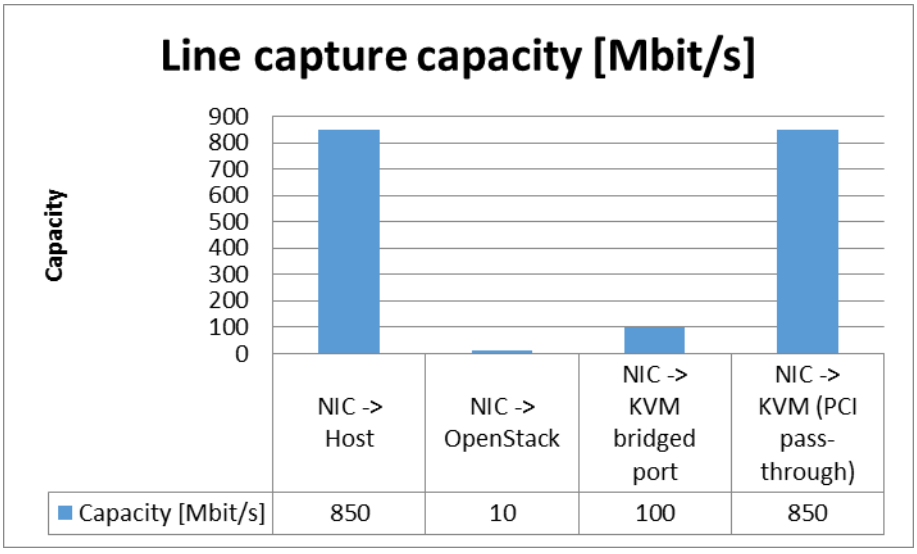

Figure 14: Line capture capacity

As we can see from the Figure, the bottleneck in the capture capacity lies between KVM hypervisor and OpenStack. Obviously, it is the Open vSwitch (OVS) sitting between KVM hypervisor and OpenStack that limits the system's capture capacity.

3.1.3.2 UDP and TCP bandwidth

In the second test case we used *iperf* –tool [*https://iperf.fr/*] to measure the bandwidth of UDP and TCP traffic over the internal interfaces of the validation system. The results of the measurements are in Figure 15 and Figure 16.

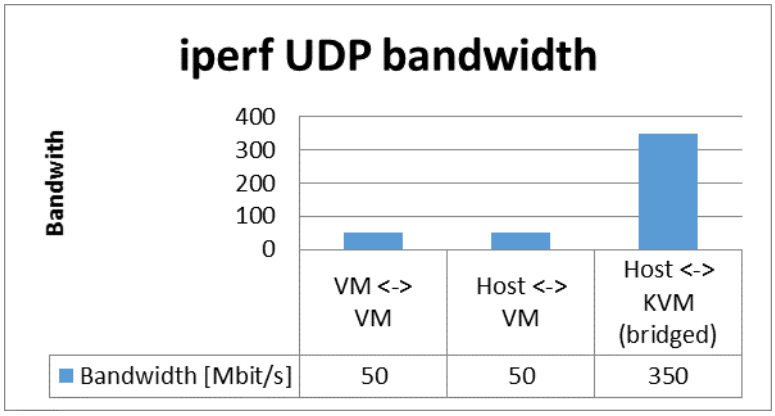

Figure 15: UDP bandwidth measurement

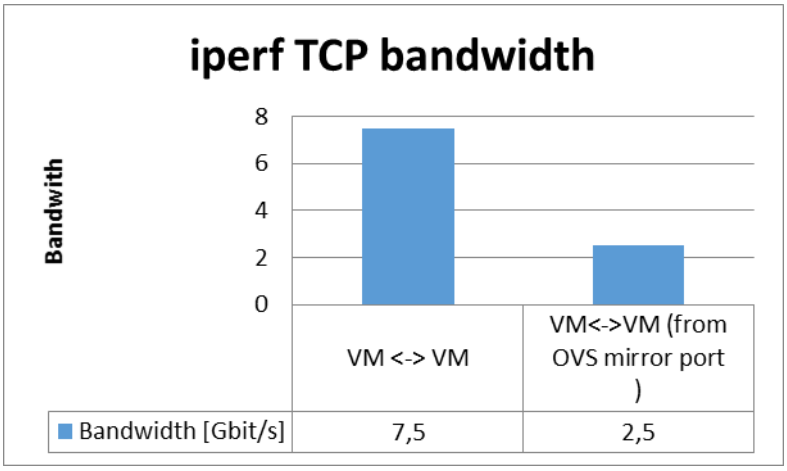

Figure 16: TCP bandwidth measurements

As we can see, there is a tremendous difference (Gbit/s vs. Mbit/s) between the measured bandwidths of TCP and UDP protocols in VM to VM communication. The root cause of this anomaly is left for further study.

As was already seen in the capture capacity tests, it seems that OVS is the bottleneck in Host-VM communications.

Another finding was that using OVS mirror to monitoring VM-VM traffic did not have a notable effect with UDP, but with the TCP protocol the bandwidth dropped to 1/3 from the original (Figure 16). Obviously, with UDP, the limitations of the mirror port are minor compared to some other limiting factors in OVS, which determine the overall (small) bandwidth with UDP.

As the system seemed to be better optimised for TCP protocol, we made yet another measurement to see how different system interfaces of the PoC affect TCP bandwidth. The results of the measurements are given in Figure 17.

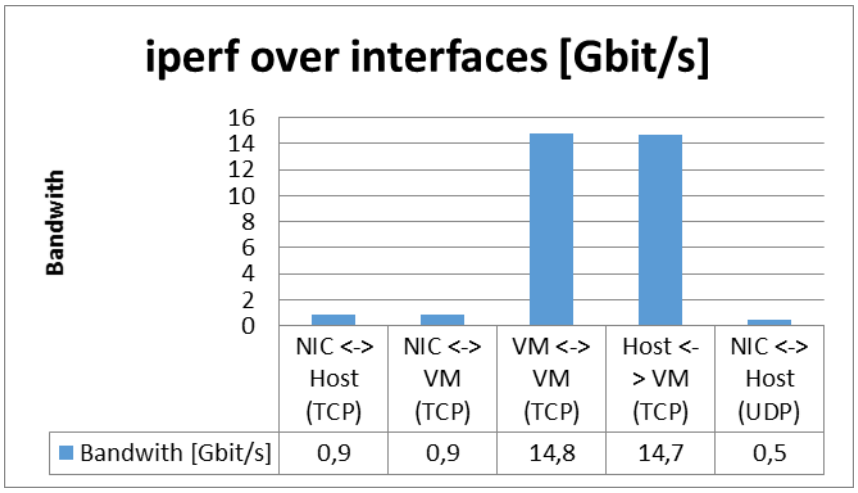

Figure 17: TCP bandwidth vs interfaces

As we see from the Figure 15 and Figure 17, the bandwidth achieved between VM-VM and Host-VM communications is roughly the same with the same (UDP or TCP) protocol. There is a huge difference between TCP and UDP bandwidths, though. Thus, it can be assumed that the component which is limiting the bandwidth must be on both communication paths (VM-VM and Host-VM). We assume OVS to be the limiting factor. Some further measurements would be needed to confirm this assumption.

3.1.3.3 Scalability

In order to test how PoC system was able to offer the increased processing capacity to VNFs, we measured the offline (=data read from a .pcap file) analysis capacity of our vAnalyzer appliance. The KVM hypervisor was configured to use 8 Host CPUs to run OpenStack environment. The scalability of the NFV system was tested by configuring the vAnalyzer to use 1, 4 and 8 finally CPUs in consecutive test runs. In each test, the analysis capacity of vAnalyzer was measured and recorded. Note that the virtual execution environment of the vAnalyzer VNF was kept the same in all test cases (8 VCPUs, 32 GB RAM, 200 GB disk space).

As a reference, we made the same measurements with a 'real' analyzer (same software version) running on (the same) Host computer. The results of both measurements are shown in Figure 18.

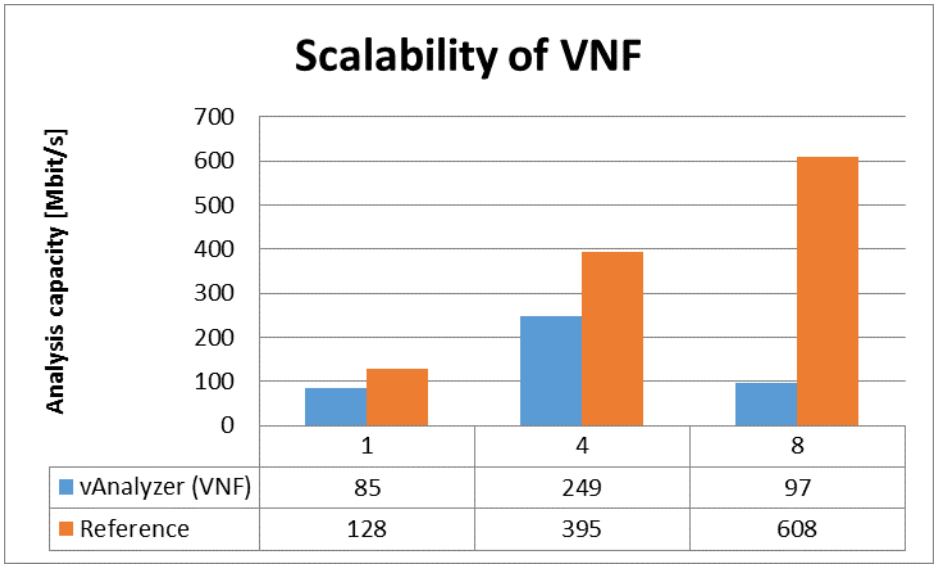

Figure 18: Scalability

As we can see from Figure 18, the NFV environment pretty much failed to utilise the increased CPU capacity. The performance of vAnalyzer with 1 and 4 CPUs was about 60 % of the reference system. Increasing the amount of CPUs to 8 dropped the performance back to original level, while the performance of the reference system increased linearly. The root cause for the degraded performance of VNF is so far unknown and needs further testing and investigations.

3.1.4 Summary

As a summary of the validation measurements, the following conclusions can be drawn:

- OpenStack as NFV environment is complicated and has a steep learning curve. It is also hard to configure and administrate.
- Based on the tests, hypervisor based virtualisation does not seem to scale too well. As an alternative, container based virtualisation technology (e.g. Docker) would be worth investigating.
- OpenStack environment is currently lacking a TAP device, which makes traffic monitoring complicated.
- The validation system's UDP bandwidth was inferior compared to TCP. It may be a configuration problem. Further testing and investigations are needed to resolve the problem.
- In the validation environment, the performance of the vAnalyzer NFV appliance was roughly 60 % of its 'real-world' counterpart. The performance penalty of the virtualisation is too high at the moment. Profiling and a detailed analysis of the performance bottlenecks are needed. Based on the measurements, OVS is currently the main suspect for the performance drop.

3.2 MMT for SDMN

Montimage Monitoring Tool (MMT) is a global monitoring and security solution that provides advanced monitoring to audit QoS and performance, and use this data to trigger security alarms and countermeasures, so enhancing the user experience by ensuring that both Performance and Security Policies and SLA terms are always fulfilled.

3.2.1 Validation objectives

Performance and security management and monitoring under the scope of SIGMONA demonstrates what techniques can be applied to manage security in SDN, NFV, and cloud environments. Montimage's work in WP2 and WP4 is described, where a high-level description of the validation platform is presented. This platform serves to validate some aspects deemed important in the QoS and security monitoring of SDMN, virtual networks and virtualised functions. In particular, through this setup, Montimage investigates where the monitoring probes or agents should be deployed and how the SDM CTRL should interact with the SDN CTRL to manage and control active or passive probes deployed for performing analysis, and trigger prevention and mitigations strategies. Likewise, Montimage's research work investigates on how at least part of the security monitoring tasks can be virtualized; how the traffic or extracted meta-data can be analysed by a probe running in the cloud; and, how the monitoring can analyse virtual connections and signalling to detect abnormal behaviour.

This validation aimed at testing the following use cases:

- The virtualisation of DPI-type monitoring for performance purposes;
- The procedure to configure and deploy this function;
- The possibility of analysing data links for evaluating QoS and estimating impact on QoE.

The validation was expected to give insights into the following issues:

- The effectiveness and cost of virtualising a DPI-type monitoring function
- Visibility of network monitoring when introducing virtualised networks and elements

3.2.1.1 Key success criteria and performance indicators

Objectives	KPI (qualitative and quantitative indicators)
Improvement of scalability, performance, costs, managing QoS/QoE, in the case of network monitoring adapted to network virtualization.	Measure scalability, in terms of cost and performance, of monitoring video transmission for analysing QoS/QoE. Compare monitoring in virtual and physical scenarios. Quantitative analysis: Measure resources needed to monitor video transmissions in different bandwidths settings (e.g., 100M, 1G, 10G). Measure any loss of precision due to loss of information. The following key performance and quality indicators need to be measured to evaluate the experimental prototype: • Maximum bandwidth without loss of information will be determined • Latency introduced will measured • Resources required (e.g., number of cores and threads needed, CPU usage, RAM memory needed, disk space needed, traffic split and aggregation required) and the cost of the solution will be estimated • Resiliency to traffic peaks will be evaluated. Scalability graph : Functionality : different levels of analysis and methods used; Cost : estimated cost of CPU/Memory/HW needed and deployment/operation efforts; Performance: resources needed wrt functionality and timeliness of detections.
Troubleshooting and detecting performance problems	Compare monitoring in virtual and physical scenarios in the detection and localization of network performance problems. Qualitative analysis: Determine the flexibility and effectiveness to detect and locate problems.
Correlate data from different sources (physical and virtual)	Determine the advantages (if any) in correlating metadata captured from the physical and virtual equipment and functions. Qualitative analysis: Determine advantages in using metadata from different sources.
Maintainability: Monitoring of the control and data planes enables the maintainability, diagnosis and repair of the SDMN.	Measure of the time to restoration from the last experienced failure (corrective maintenance only). Evaluate the effort to diagnose, maintain and repair the system manually versus automatically.

3.2.2 Validation environment

Note that this work is ongoing and the final results will be available by the end of March 2016 because the French consortium's contributions will end in April 2016.

To validate Montimage's monitoring solution (MMT) in the SDMN context, it is being deployed on French testbed described in [SIG1]. Figure 19 below shows the virtualised network environment.

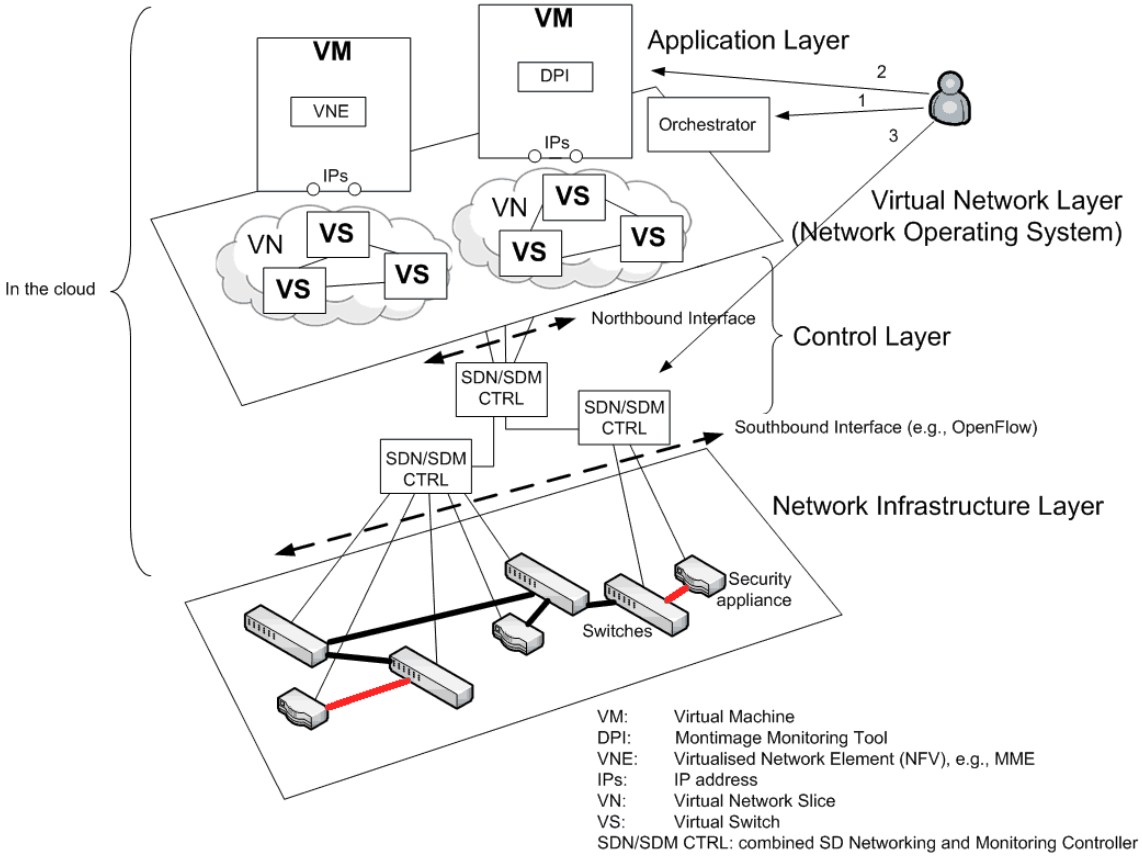

Figure 19: Network monitoring

In this figure, the user (e.g., operator) will: 1) inform the Orchestrator to deploy a virtualised DPI (vDPI) application; 2) configure this application (deploy the rules or properties that need to be detected); and, 3) inform the SDN controller to direct the network traffic to be analysed to the vDPI function. How this procedure can be automated to reduce the interventions that a human operator needs to perform will be also studied.

Deployed rules will be specified to analyse and detect performance properties of the virtualised and physical connections, allow detecting and locating performance problems, and help verify that SLAs are respected.

3.2.3 Validation scenarios, methods and results

Figure 19 shows a conceptual deployment of the Montimage Monitoring Tool (MMT) in an SDN environment in which MMT probes are located on a virtual machine or could be co-located in the virtual machines with the network functions.

Figure 20 shows how the different monitoring components need to be deployed in an NFV architecture. As shown, probes can be deployed in the VNF or as a VNF, as well as in the Virtual Infrastructure Manager, alongside Celiometer, to detect security related events. To deploy these probes, manage them, and obtain a complete picture, monitoring components or applications need to be deployed in the VNF Manager and the Network management system (OSS/BSS).

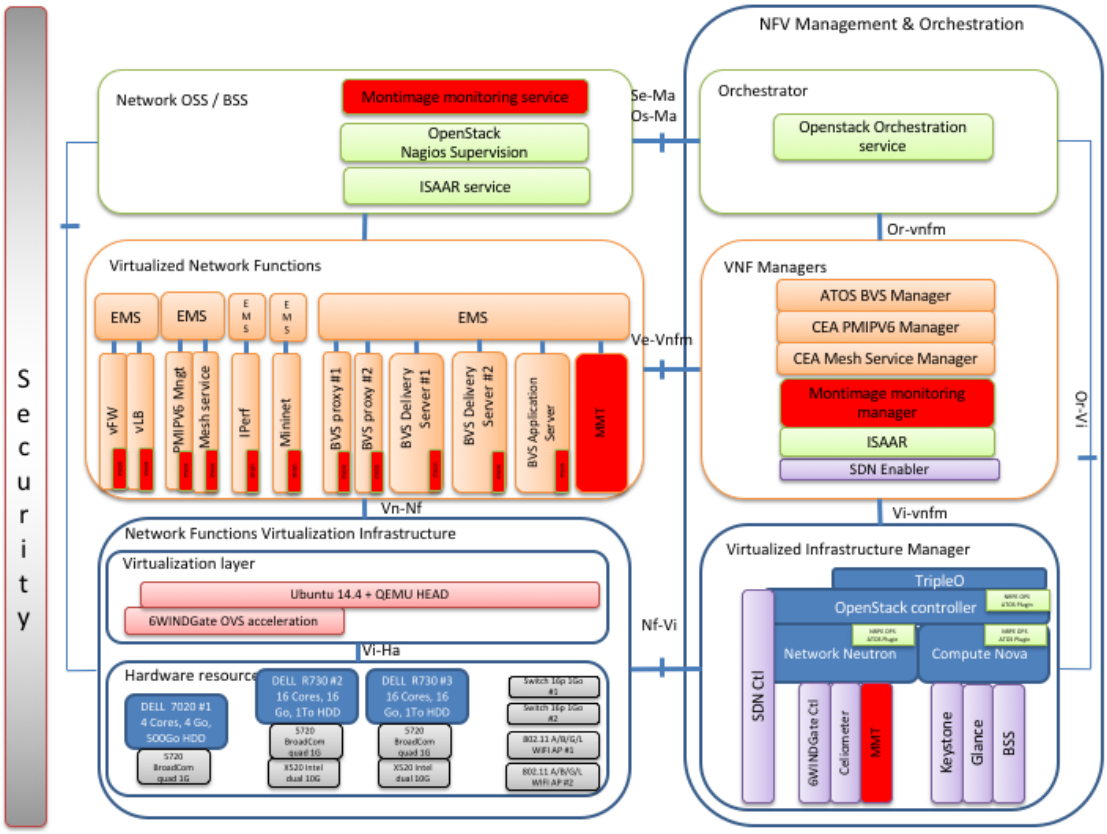

Figure 20: Deployment of monitoring functions in the NFV architecture

3.2.3.1 On the deployment of the monitoring functions

NFV introduces virtualized networks and functions that need to be monitored. To be able to assure QoS and end-user QoE, a monitoring architecture needs to be defined and deployed in order to measure and analyse the network flows at different observation points that could include any component of the system, such as physical and virtual machines. Setting up several observation points will help to better diagnose the problems detected. With SDN, it is possible to create network monitoring applications that collect information and make decisions based on a network-wide holistic view. This enables centralized event correlation on the network controller, and allows new ways of mitigating network security breaches and faults.

The monitoring probes can be deployed in different points of the system. Let us consider a single hardware entity that is controlled by a hypervisor that manages the virtual machines. A first approach consists of installing the monitoring solution (MMT) in the host system (hypervisor) that operates and administers the virtual machines (see Figure 21), in this way providing a global view of the whole system. This approach requires less processing power and memory to perform the monitoring operations, since the performance assurance is located in a central point. In this way, network connections between the host and the virtual machines can be easily tracked allowing early detection of any performance issue. The main problem of this approach resides in the minimum visibility that the host machine has inside the virtual machines, not being able to access to key parameters such as the internal state, the intercommunication between virtual machines, or the memory content.

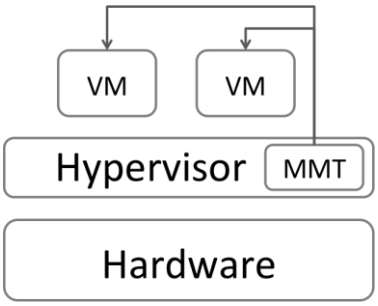

Figure 21: Network-based protection

Monitoring probes can also be located in a single privileged virtual machine that is responsible for inspection and monitoring of the rest. This approach is called Virtual Machine Introspection (VMI). It offers good performance since the monitoring function is co-located on the same machine as the host it is monitoring and leverages a virtual machine monitor to isolate it from the monitored host. In this way, the activity of the host is analysed by directly observing hardware state and inferring software state based on a priori knowledge of its structure. VMI allows the monitoring function to maintain high levels of visibility, and even enables the manipulation of the state of virtual machines. Unfortunately, VMI based monitoring software is highly dependent on the particular deployment and requires privileged access that cloud providers need to authorize.

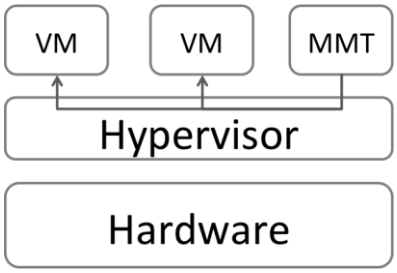

Figure 22: Virtual machine introspection

The approach that offers the best solution in terms of functionality and feasibility is the deployment of the monitoring tools in every virtual machine. In this way, robust assessment can be achieved since the monitoring software has a complete view of the internal state of every virtual machine, as well as the interactions with the host or any other virtual machine. Figure 23 shows how this approach can be deployed.

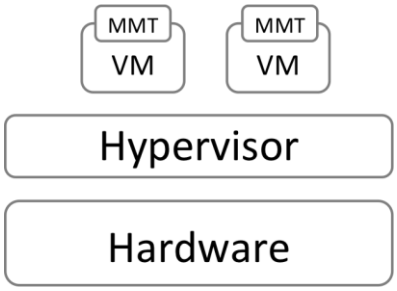

Figure 23: Host-based protection

This third solution offers the best performance since the probes (or agents) can be configured to extract only the minimum set of information required for assessing the performance, and the processing power and memory required are distributed among the virtual machines. Furthermore, its deployment is simpler than other approaches since it can be included in the software image of the virtual machine, so it is automatically initiated when instantiating each virtual machine with no further configuration needed.

Despite of the individual probes installed on each virtual machine, there is the need of a global monitoring coordinator that supervises the monitoring tasks of each probe installed on each virtual machine. For this, each probe must be able to directly interact with any other probe (i.e., peer-to-peer interactions), as well as with the monitoring coordinator. Local decisions can be taken by the individual

monitoring probes installed on each virtual machine, and the monitoring coordinator can perform coordination, orchestration and complex event detection.

The KPIs will be measured to give some indication on how the monitoring solution can be made more efficient with respect to OPEX and CAPEX. In particular, the scenarios previously described will provide results that can be used to improve both the performance and security of SDMN by: analysing information at different levels (physical, virtual and OSI layers 2 to 7); showing how the network appliances and services can be managed to improve the efficiency of the monitoring; how performance and security policies and filtering rules can be deployed and enforced; and, showing how cloud computing can be used to improve the flexibility and scalability of the monitoring solutions.

3.2.3.2 Summary

Considering the different monitoring deployments presented in the previous section, herein, a whole architecture integrating monitoring probes and coordinator is presented.

Figure 24 represents a possible deployment scenario for MMT in an SDN environment. As depicted, MMT probes capture performance and security meta-data from each virtual machine, and are able to perform countermeasures to mitigate performance and security problems. MMT probes have the capacity of peer-to-peer communication, so they can share relevant information with the aim of increasing the efficiency of the mechanisms and, thus, ensure the correct operation of the whole system. To perform coordination and orchestration of the whole monitoring system, a central MMT Operator receives the information from the distributed MMT probes. The MMT Operator is also in charge of correlating events to create reports to inform network managers of the system activities and countermeasures taken. Furthermore, it will be able to globally analyse the information provided by individual MMT probes with the ultimate objective of detecting complex situations that may compromise the system.

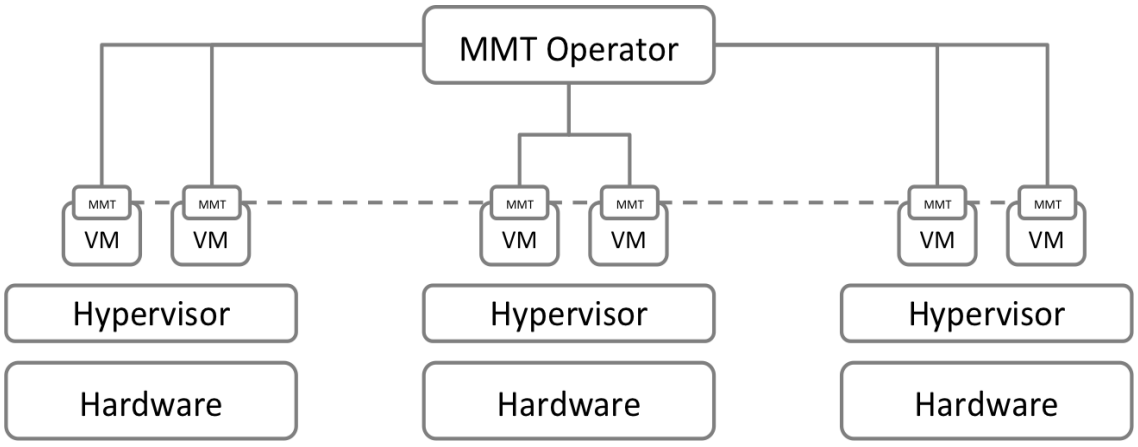

Figure 24: MMT architecture deployment for SDN

4. Conclusions

This document collects the results of the validations of the SDMN concepts and solutions proposed in WP2. The validations pertain to a wide range of research and development topics in the areas of

- network control and management:
  - hierarchical management and control of SDN-enabled mobile operator infrastructure (base stations and backhaul)
  - mobile backhaul traffic management and network performance optimization with SDN
  - virtualized hybrid-SDN backhaul network in multi-operator environments
  - secure backhauling
- monitoring
  - adaptation of a control plane protocol analyzer to virtual environments – functionality and performance
  - DPI monitor virtualization in SDNs – automated configuration and QoS/QoE measurements

For each proposed solution, the validation objectives with related key success and performance criteria; and validation environments, methods and results have been presented. Some of the validations are still ongoing at the time of writing this report. The following conclusions can be made for the validations for which final or at least preliminary results are already available:

In *Integrated RAN/SDN controller* validation (section 2.1), a controller hierarchy was used for managing and controlling SDN-enabled mobile network infrastructure. A load-balancing algorithm was devised for balancing the load of flow requests, originated from the SDN switches, among the controllers. The algorithm proved to outperform baseline algorithms that applied random reassignment of flows to controllers or reassignments to the topologically nearest controller.

In *Mobile backhaul management and optimization* validation (section 2.2), a real LTE network with SDN mobile backhaul was built in order to show that by combining the SON for MBH solution and SDN technology, it is possible to create a self-configuring and self-optimizing mobile backhaul that can continuously adapt itself to the actual traffic demand. The validation work has shown that such operation is possible to achieve and it can introduce ~20% CAPEX reduction.

In *Virtualized hybrid-SDN mobile backhaul* validation (section 2.3), virtualization techniques were used to abstract a Wireless Mesh Network (WMN) to a virtual SDN switch in a small-cell backhaul that was sliceable among different MNOs. The usage scenario involved auto-enrolment of a MNO's base station, which, after authentication, becomes visible in the MNO's network slice. The SDN concept with its explicit controllability of the traffic flows appeared to serve well in the scenario that required authenticated network attachment and subsequent on-demand connection provisioning. Similarly, the abstraction and virtualization concepts and tools allowed effective hiding of the idiosyncrasies of the WMN and enabled network slicing among MNOs. On the other hand, data plane resiliency seems still to be best achievable by exploiting legacy link aggregation features that are available in hybrid Ethernet/SDN switches.

In *Secure backhauling* validation (section 2.4), COTS servers were applied for securing communications between small cells and network aggregation points with the aim of verifying whether the servers have enough encryption and routing capacity to sustain surging mobile bandwidth. The test proved that the performance of the COTS servers was comparable to high-end dedicated hardware. Further, a cost study showed that the cost per encrypted IPSec Gbps of COTS based solutions can be a fifth of the cost of dedicated equipment.

In *Virtual analyzer* validation (section 3.1), a live network analyzer was applied in a virtual machine in the OpenStack environment. The aim of the validation was to assess the data capture capacity and packet trace analysis performance. In addition, the UDP and TCP bandwidths between virtual machines were measured. The performance of the virtual analyzer was roughly 60 % of its 'real-world' counterpart. This indicates high performance penalty of the virtualisation, which was exacerbated by poor scalability with respect to the number of used CPU cores. In addition, OpenStack is currently lacking a TAP device, which makes traffic monitoring complicated. Furthermore, for an unknown reason the validation system's UDP bandwidth was inferior compared to TCP.

In *MMT for SDMN* validation (section 3.2), Montimage's monitoring solution (MMT) has been applied in the SDMN context, with the aim at investigating the impact of virtualisation in the performance of DPI-type monitoring, configuration and deployment options, and QoS/QoE evaluation. So far, the validations have focused on identifying the most feasible location for the monitoring probes in the virtualised

network infrastructure. The probes may be deployed in the host system (hypervisor), in a single privileged virtual machine, or in every virtual machine. The last option seems the most feasible and gives the best performance. Still, this mode of deployment requires a centralized monitoring coordinator in the network that supervises the monitoring tasks of each probe.

5. References

[CEN] Centec V350 Series Switch, [ONLINE], Available: http://www.centecnetworks.com/en/SolutionList.asp?ID=43

[CO] Coriant 8600 Smart Router Series, [ONLINE], Available: http://www.coriant.com/products/8600.asp

[FB] Nokia Flexi BTS, [ONLINE], Available: http://networks.nokia.com/portfolio/products/mobile-broadband/single-ran-advanced/flexi-multiradio-10-base-station

[FG] Nokia Flexi Network Gateway, [ONLINE], Available: http://networks.nokia.com/portfolio/products/evolved-packet-core/flexi-network-gateway

[FLO] FlowVisor, [ONLINE], Available: https://github.com/OPENNETWORKINGLAB/flowvisor/wiki

[IND] Indigo Virtual Switch, Project Floodlight, [ONLINE], Available: http://www.projectfloodlight.org/indigo-virtual-switch/

[MIN] An Instant Virtual Network on your Laptop (or other PC), [ONLINE], Available: http://mininet.org/

[NS3] Network Simulator 3, [ONLINE], Available: http://www.nsnam.org/

[OD] OpenDaylight, [ONLINE], Available: http://www.opendaylight.org/

[OF13] ONF, "OpenFlow Switch Specification, Version 1.3," 25 June 2012. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf.

[OVS] Open vSwitch, An Open Virtual Switch, [ONLINE], Available: http://openvswitch.org/

[OVX] OpenVirteX, Programmable virtual networks [ONLINE], Available: http://ovx.onlab.us/

[RYU] Ryu, a component-based software defined networking framework [ONLINE], Available: http://osrg.github.io/ryu/

[SIG1] Deliverable D1.2: “2nd Consolidated view of SIGMONA Software Defined Mobile Network Architecture”, Section 4.3.1: “QoS, mobility and SW accelerated architecture Mapping”; SIGMONA project, 2016

[SIG2] Deliverable D2.1: “Control and management of mobile and transport networks in SDMN”, SIGMONA project, 2016