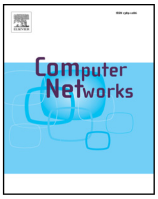
Computer Networks 129 (2017) 142–158

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Implementation experience in multi-domain SDN: Challenges, consolidation and future directions

K. Katsalis[a,*], B. Rofoee[b], G. Landi[c], J.F. Riera[d], K. Kousias[e], M. Anastasopoulos[b], L. Kiraly[f], A. Tzanakaki[b], T. Korakis[e]

[a] *Eurecom, Mobile Communications, Biot, France*
[b] *University of Bristol, UK*
[c] *Nextworks, Italy*
[d] *Fundacio I2CAT, Distributed Applications and Networks, Barcelona, Spain*
[e] *University of Thessaly, Dept. of ECE, Greece*
[f] *Juniper Networks, USA*

ARTICLE INFO

Article history:
Received 8 July 2016
Revised 10 September 2017
Accepted 11 September 2017
Available online 22 September 2017

Keywords:
SDN
Cloud networking
Wireless
Optical
NFV
Testbed experimentation
Network integration
LTE
Wi-Fi

ABSTRACT

Network architectures compliant with the Software Defined Networking (SDN) design paradigm, are expected to provide extreme flexibility for service orientation and allow for efficient use of network resources of cloud systems. Nevertheless, radical reconsidering and removal of boundaries set out when studying multi-domain communications are required, in order to unleash the hidden potential of SDN and provide a "holistic" network view. Although many domain-specific efforts have been proposed in the literature and indeed they gain a lot of industrial attention, real multi-domain SDN implementations over converged wireless-optical networks are just starting to disclose.

In this work we present an open end-to-end multi-domain SDN system, while focusing on the necessary abstractions and virtualization techniques to integrate virtual wireless and optical resources. With the proposed system, called CONTENT, we shed light in the field of wireless-optical network virtualization from an end-to-end perspective. We present the architecture and the integrated testbed that realize the envisioned system. Evaluation results are provided and implementation experience is reported using the integrated solution. The way SDN methodologies and techniques can be used to support NFV concepts in end-to-end fashion are also presented.

1. Introduction

With the propensity for ubiquitous software defined networking in all network segments, an immersive virtual networking environment promises to debunk all the challenges encountered in network configuration. Some of these challenges could be briefly described. Management and control information needs to be exchanged across multiple and geographically distributed network domains, causing increased service setup and state convergence latencies [56]. In such highly complex environments the probability that management/control data is lost (e.g., information related to round trip time delays, packet loss rates, transmission rates etc.) is high, often leading to violation of quality of service (QoS) requirements. In addition, increase in the network size makes the monitoring needed for the collection of the control data and the management of all network elements impractical and inefficient. Furthermore, in highly dynamic environments, services requests can have big and rapid variations that are not known in advance.

In order to manage, control and efficiently operate this type of complex infrastructures in an efficient manner, Software Defined Networking (SDN) [15,40] has been recently proposed as a key enabling technology. In SDN, the control plane is decoupled from the data plane and is moved to a logically centralized controller that has a holistic view of the network. However, although SDN technologies promise to leapfrog an entire generation of technology and even if we are able to build virtual wireless or virtual optical networks, the truth is that the challenge of integration in end-to-end scenarios, still remains.

In this study, we present a control and management framework called CONTENT (Converged Wireless Optical Network and IT Resources), adopting the concept of multi-domain network virtualization. Its objective is to integrate virtual wireless and optical resources, through the development of novel virtualization and network softwarization techniques based on SDN. In the wireless domain a hybrid Wi-Fi/ LTE network has been considered, whereas

[*] Corresponding author.
E-mail address: kostas.katsalis@eurecom.fr (K. Katsalis).

https://doi.org/10.1016/j.comnet.2017.09.005
1389-1286/© 2017 Elsevier B.V. All rights reserved.

in the optical segment a frame-based optical network solution (known as TSON) has been adopted [57]. We present a fully operational multi-domain SDN system that: (a) considers virtualization of both the wireless and optical resources; (b) exposes virtual resources using a common virtualization framework; (c) facilitates the access to remote virtual network resources by Mobile Optical Virtual Network Operators (MOVNOs); (d) confronts the necessary SDN mechanisms to handle virtual wireless and optical segments seamlessly in the control plane; (e) allows for service orchestration above the control layer responsible for the instantiation of end-to-end cloud services for mobile users and their management at run-time and (f) supports service provisioning through Virtual Network Functions (VNFs).

The contributions of this paper are the following:

- We describe the architectural requirements and the motivation for end-to-end converged solutions and give a note on state of the art.
- We provide a functional description together with a detailed structural presentation of a proposed end-to-end architecture.
- We elaborate the virtual resources specification, provisioning, management and operation mechanisms used in the integrated system.
- We present the software components and the interfaces responsible for within data plane and control plane functionalities per domain, while also for the interaction between the Data-plane with the control plane of the architecture.
- We present an extended SDN/NFV testbed provided by Juniper Networks, where different SDN controllers are used to control different segments of the network.
- We describe the way the proposed system is able to support NFV concepts and service chaining.
- we present performance evaluation results on all the layers of the architecture.

In our system the virtualized wireless domain, is connected through virtualized optical links with the remote (virtualized) data-center. An Infrastructure Management Layer is introduced for the management of the virtualized networks, while an integrated SDN control approach is used over the abstracted and virtualized physical underlay. Although there are many works recently that are trying to explore the use of SDN technologies especially in the wireless domain (like [17,24,34]), the proposed approach facilitates control and management of vertical network slices on per Mobile Optical Virtual Network Operator (MOVNO) basis.

Note that the concept of wireless-optical integration is gaining significant attention because of the realization of concepts like the Cloud-RAN [13,35]. In this concept optical networks are used to support the backhaul and fronthaul links, in the ”virtualized” Radio Access networks (RAN), where functional splitting of the BBU and RRH functions is performed. In true Cloud-RAN based production systems, the wireless domain network and the optical domain network must be physically collocated. In our results presented, the GEANT network [1] sits between the two domains, imposing extra network delays in the realization of the Cloud-RAN system; nevertheless the results and the approach we present in this work still rely on a real end-to-end system, while the vertical interfaces build can also be exploited by other Cloud-RAN designs and systems.

To the best of our knowledge, the proposed architecture, the vertical interfaces built and their implementation between the physical underlay, a virtualization layer and the SDN control plane, are among the first to presented and evaluated in a true end-to-end setup. Our design can be adopted and work in parallel with LTE/Wi-Fi specific SDN mechanisms, since the control in our case runs on top of the virtualized infrastructure. This paper can serve as a guide for other wireless access and optical network providers and testbeds, interested in supporting the SDN/NFV design paradigms over multi-domain network infrastructures.

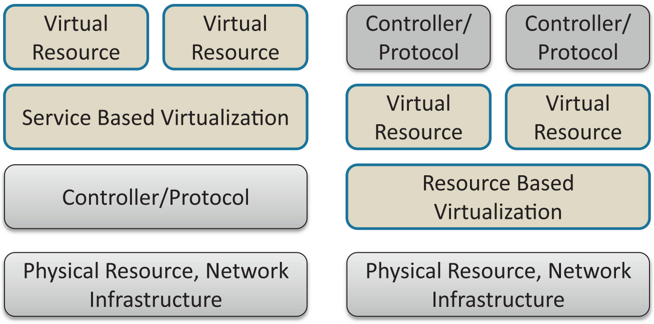

Virtual Resource
Virtual Resource
Service Based Virtualization
Controller/Protocol
Physical Resource, Network Infrastructure
Controller/ Protocol
Controller/ Protocol
Virtual Resource
Virtual Resource
Resource Based Virtualization
Physical Resource, Network Infrastructure

Fig. 1. Service (left) versus resource (right) virtualization models.

The rest of the paper is organized as follows. In Section 2 we provide background information regarding end-to-end virtualization. In Section 3 we describe the proposed system and network architecture. In Section 4 we elaborate the components involved for the data plane and control plane integration and we describe the implementation aspects of the layered architecture. In Section 5 we present the performance evaluation of the proposed solution, using real exercises on the integrated testbed. In Section 6 we describe two cases studies related to NFV experiments for parental control and service differentiation in the context of MOVNOs. In Section 7 we highlight a number of challenges we had to consider, open issues and planning of our future work. We conclude the paper in Section 8.

2. Related work on multidomain SDN

Towards 5G communications denser deployment of access nodes move a number of challenges from the access interface to the backhaul network. The reason is that because of the tremendous increase in mobile traffic and the pressure to the operators, efficient integration of wireless, fiber and possibly other transport solutions has become a necessity. Although backhaul links can be supported by technologies like 60 GHz, transport solutions over fiber seems to be the preferred approach when it comes in designs with large capacity requirements.

Note that efficient wireless-optical network abstraction, virtualization, management and control is required in various network setups and concepts. Either we are discussing about the Cloud-RAN concept or traditional multi-domain networking, the SDN approach offers the potential to exercise direct control from the control plane over the state in the networks data-plane elements, via a well-defined APIs. The requirement to apply SDN control is that the physical underlay is programmable (Fig. 1).

Although the SDN concept seems to have appeared suddenly, and the technology has still not reached a point of maturity, plenty of research activities exist that describe the concept, while a satisfactory number of works provide within domain procedures optimization. Indeed, in order to extend the business models of the involved stakeholders, very active research is performed on exploiting the resource and service virtualization models.

2.1. SDN concept

The concept along with an extended related work survey on SDN are presented in various works like [15,32,40]. In [32,40] the key building blocks of an SDN infrastructure are discussed, while in-depth analysis of the hardware infrastructure, southbound and northbound APIs, network virtualization layers, network operating systems (SDN controllers), network programming languages, and network applications are also presented. In [15], the distinction between service and resource based virtualization is presented, that is also the one we also consider.

Resource-based virtualization (or resource virtualization) becomes the most suitable option in order to have the finest feasible granularity in any case, as well as the maximum level of flexibility to control the different virtual resources. However, in terms of implementation, the complexity of this approach increases, since it needs to completely control the physical resource itself. On the other hand, service-based virtualization loses part of the flexibility, since the virtualization system is located on top of the network service. Service-based virtualization implies that the virtualization intelligence does not lay directly over the resource, but over the control plane deployed directly over the physical resource. As an example, the L3VPN, should be an example of service-based virtualization, since the virtualization happens of top of the IP protocol.

As we will describe, in our approach the virtual infrastructure operates over the wireless domain using the service virtualization based approach, while in the optical domain we utilize the resource-based virtualization approach. The architecture we propose is open and flexible to adopt both. More recently, in [58] a Stateful Data Plane Architecture (SDPA) was proposed to overcome the absence of stateful forwarding functionality in the OpenFlow protocol, through a co-processing unit, called Forwarding Processor, that manages state information through new instructions and state tables. Panda et al. [41] address the important of identifying the consistency models for coordinating the actions of a replicated set of SDN controllers, which is very important for the multi-domain design that we consider.

2.2. SDN in the wireless domain

Wireless virtualization and SDN control in the wireless domain are presented in works like [7,17,24,26,34,55]. For example in [55] an architecture for software-defined RAN via virtualization called OpenRAN is presented, while in [7] the CROWD solution is presented for SDN control in DenseNets. More recently in [47] an architecture for HetNets that incorporates various elements of SDN and software defined wireless networking. A taxonomy of mobile networks following an SDN/NFV approach is presented in [39,43]. For the LTE network in [48] the use of OpenFlow is presented inside the EPC network, IP-based routing in virtualization-based LTE EPC architecture (vEPC) is considered in [38]. In [42] an OpenFlow SDN framework for Wi-Fi networks is proposed. An SDN-based plastic architecture for 5G networks consisting of unified control plane and a clean-slate forwarding plane is presented in [50]. In [6] the SoftAir solution is presented for seamless incorporation of Openflow, mobility-aware control traffic balancing, resource-efficient network virtualization, and distributed and collaborative traffic classification towards 5G wireless communications.

2.3. SDN in the optical domain

Regarding optical network programmability, the majority of the ongoing research efforts have been focused on OpenFlow based SDN implementations [12] with the objective to apply SDN concepts across multilayer multivendor networks in supporting a unified control structure [23]. Although the current OpenFlow protocol is originated from packet switching (i.e., Ethernet), it is also considered as a realistic unified control plane solution for integration of packet and optical circuit switched networks [18]. However, its main limitation is that it does not support optical network features like switching constraints and optical impairments. Furthermore, OpenFlow does not support advanced and emerging optical transport technologies, such as a flexible Dense Wavelength Division Multiplexing (DWDM) grid. To address the shortcomings of the current OpenFlow extensions on supporting optical network technologies, the authors in [12] proposed a generic and extended optical flow specification.

2.4. Multi-domain SDN architectures and activities towards 5G

In the field of wireless-optical multi-domain networking, great research activities exist towards integrated 5G communications. In the following we present an indicative set that is most relevant to our work. Multidomain virtualization has been investigated in [9,49]. The application of the SDN approach in the Telecom domain is presented in [27], while SDN application in the backhaul is the focus of the work presented in [17]. Multi-tier cellular systems are presented in [26,28,53] and multi-domain modeling work considering both the wireless and optical domains was presented in [51]. The way SDN is expected to play role n the 5G evolution is present in [4] and the METIS project, the corner stone of the 5G-PPP initiative activities.

Note that RAN densification [7,45] and Cloud-RAN (C-RAN) [13,35] seem the most promising candidates to meet the extreme requirements of 5G communications and the exponential increase in user traffic. A number of research activities and projects investigate wireless-optical integration under the C-RAN concept and backhaul/fronthaul wireless-optical integration [45], like 5G-XHaul [25] and the 5G-CrossHaul [19] solutions. The main idea behind C-RAN is to pool the Baseband Units (BBUs) that are responsible for the processing of IQ data, from multiple base stations into centralized BBU Pools and connect thousands of remote Remote Radio Heads (RRH) to these centralized BBU pools. The burden in this approach is shifted to the high-speed wire-line transmission of IQ data. An overview of the C-RAN technologies are described in [13]; while [11] analyzes the packetization and packet scheduling impact on different functional splits considering the packet-based transportation. Fiorani et al. [21] analyzes the energy performance of four radio access network (RAN) architectures with optical transport, each one utilizing a different option for splitting the base-band processing functions. Chanclou et al. [10] describe a WDM fronthaul network with passive monitoring at the antenna site and automatic wavelength assignments. An alternative approach considers for Ethernet based solutions. For example Gomes et al. [22] proposes to utilize Ethernet as the underlying transport layer with low-latency Ethernet switching for the RRU-BBU functional split;Venmani et al. [52] provides a synchronization architecture for Ethernet-based fronthaul interface.

In the approach we describe in the following sections, the same vertical interfaces design and implementation can be applied for both the wireless and optical segments as in the case of C-RAN. For example Gutiérrez et al. [25] and De La Oliva et al. [19] adopt a similar approach like ours on the way virtual resources are exposed and controlled by a unified control plane.

3. Layered architecture overview

Our goal was to provide a framework that is able to support cloud services, provide automation support in processes which involve different stakeholders, while it is easy to be deployed by existing wireless and optical network technologies. How can virtual networks be created in all the domains? Which mechanisms are required to perform SDN control in all network segments and integrate the optical and wireless domains, in both the data and control planes of the architecture? How network slices are created per virtual network operator? These are the basic challenges our solution addresses, exploiting a multi-domain architecture that is

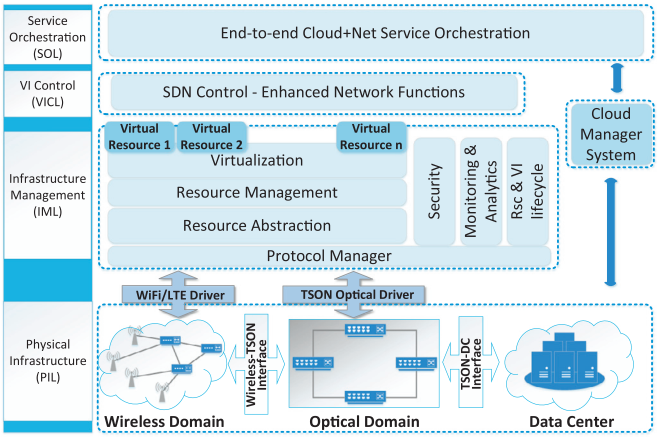


Fig. 2. The CONTENT layered architecture.

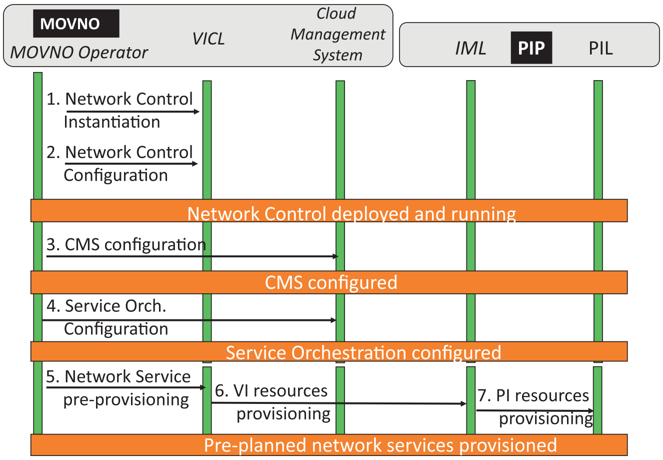

MOVNO
MOVNO Operator
VICL
Cloud Management System
IML
PIP
PIL
1. Network Control Instantiation
2. Network Control Configuration
Network Control deployed and running
3. CMS configuration
CMS configured
4. Service Orch. Configuration
Service Orchestration configured
5. Network Service pre-provisioning
6. VI resources provisioning
7. PI resources provisioning
Pre-planned network services provisioned

Fig. 3. Control Layer deployment and configuration sub-phase sequence diagram.

SDN-based, Open Networking Foundation (ONF) aligned [16], flexible enough for extreme service orientation.

3.1. The proposed layered architecture

In order to provide end-to-end network virtualization capabilities, agnostic to the underlying technological dependencies, a cross-technology architecture is proposed comprising the following layers. See Fig. 2 for a visual representation:

- *Heterogeneous Physical Infrastructure Layer (PIL):* this is related with *ONF's Infrastructure Layer*. Our focus stays on a hybrid LTE/Wi-Fi access network supported by an Openflow-based wired backhaul network, an optical metro domain, supporting frame-based sub-wavelength switching granularity and virtualized Data Center (DC) infrastructures.
- *Infrastructure Management Layer (IML):* this is related to ONF's *Control Layer*. The IML is responsible for the creation and management of virtual network infrastructures over the underlying physical resources.
- *Virtual Infrastructure Control Layer (VICL):* The VICL is responsible to provision IT and connectivity services across the different network domains and expose functionalities to the Application layer. The control is made over the IML resources (virtual or physical).
- *Service Orchestration & Application Layer (SOL):* This is related to ONF's Application Layer and is responsible to orchestrate network-based applications and coordinate the combined delivery of cloud and virtual network resources.

First note that the architecture is not bind to a specific network element, controller technology, southbound/northbound protocols or any testbeds' control framework, already deployed. This protocol independence, allows this architecture to be adopted by various wireless and optical network providers that are interested to tender and build the underlay infrastructures of MOVNOs. Note that all management, control, configuration actions, including also the reservation of the resources are mediated through the IML and involve both physical and virtual resources.

The layered architecture depicted in Fig. 2, is conceived to provide cloud and mobile cloud services on top of virtual infrastructures that span across multi-technology and multi-domain physical networks. The cooperation and interaction among the different architecture layers is described through a series of planning and operation workflows analyzed as follows:

Virtual Infrastructure(VI)Planning Phase: The VI planning phase is defined as the stage were the virtual infrastructures are requested, planned, and instantiated according to the MOVNO requirements. This phase comprehends the complete set of pre-operation actions. On one hand, it includes the different procedures where the heterogeneous physical infrastructure is registered into the IML. On the other hand, it also involves the planning and dimensioning of the different virtual infrastructures as a function of the requests coming from the different MOVNOs. Finally, the virtual infrastructure instantiation is the process that will prepare all the components within the VI in order to make it operable by the corresponding control plane. PIP stands for the Physical Infrastructure Provider.

Virtual Infrastructure (VI) Operation phase: The VI operation includes two distinct sub-phases: (a) the Control Layer deployment and configuration and (b) the cloud service operation. The former represents a pure management action performed by the MOVNO on the VI rented from the PIP; it aims at deploying and configuring an instance of the network control layer on top of the virtual network infrastructure and in some cases it may include further network service pre-provisioning actions needed to accommodate pre-planned cloud based end-to-end traffic. Moreover, to enable the integrated control and orchestration of cloud and connectivity services, specific configurations must be enforced on the Service Orchestration Layer and the Cloud Management System (CMS) responsible for the management and control of the DC resources. Fig. 3 depicts this initial deployment and configuration sub-phase sequence diagram, mainly showing the interactions among layers and actors in the architecture.

Cloud service Operations: cloud service operation is the core part of the VI operation phase, and includes all the actions needed to provide on-demand cloud services to the mobile cloud users, from the service requests, up to seamless reservation and provisioning of virtual and physical network and IT resources. This phase involves all the layers in the architecture and all the actors in the proposed ecosystem.

In the following sections we provide a comprehensive analysis regarding each layer description, implementation and performance evaluation details. This work provides a feasibility study for true end-to-end SDN integration, that can be used as benchmark against other proposals.

4. Implementation aspects of the layered architecture

4.1. Physical infrastructure layer of the end-to-end testbed

In Fig. 4 the network diagram of the integrated testbed is presented, while in the following subsections we provide domain specific details. The underlying physical infrastructure that supports

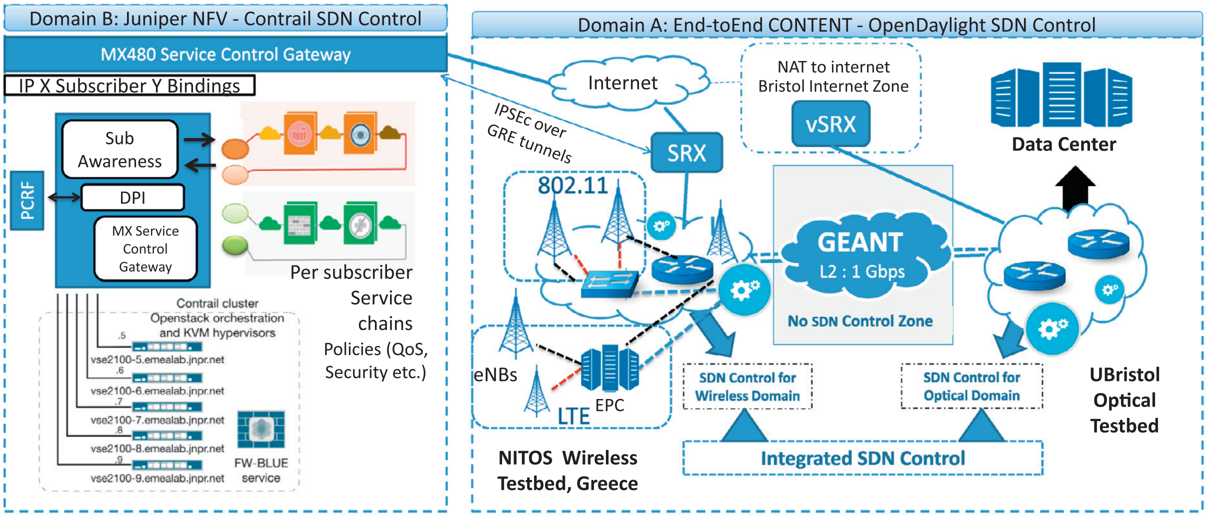

Domain B: Juniper NFV - Contrail SDN Control
MX480 Service Control Gateway
IP X Subscriber Y Bindings
Sub Awareness
PCRF
DPI
MX Service Control Gateway
Per subscriber Service chains
Policies (QoS, Security etc.)
Contrail cluster
Openstack orchestration and KVM hypervisors
.5
vse2100-5.emealab.jnpr.net
.6
vse2100-6.emealab.jnpr.net
.7
vse2100-7.emealab.jnpr.net
.8
vse2100-8.emealab.jnpr.net
.9
vse2100-9.emealab.jnpr.net
FW-BLUE service
Domain A: End-toEnd CONTENT - OpenDaylight SDN Control
Internet
NAT to internet Bristol Internet Zone
vSRX
Data Center
IPSEc over GRE tunnels
SRX
802.11
GEANT
L2 : 1 Gbps
No SDN Control Zone
eNBs
LTE
EPC
SDN Control for Wireless Domain
SDN Control for Optical Domain
UBristol Optical Testbed
NITOS Wireless Testbed, Greece
Integrated SDN Control

Fig. 4. The CONTENT Testbed: A Wi/Fi-LTE wireless testbed, with openflow wired backhaul network, interconnected to the data center through a TSON optical testbed. Per subscriber services, packet steering functions are provided through a remote Service Gateway system (left side of the figure).

the integrated multi-domain testbed comprises the NITOS wireless testbed, in Greece [2,31], interconnected through the GEANT network, to a TSON optical testbed, in UK [57]. It also considers DC infrastructure located in the Univ. of Bristol built around Openstack technology, that is used for the deployment of endpoint services and applications. This integrated testbed offered the realization of the actual end-to-end path. Furthermore, this end-to-end testbed is extended with an additional DC infrastructure that supported a NFV testbed, provided by Juniper Networks [30] and located in Amsterdam. This provided the necessary Network Function Virtualization Infrastructure (NFVI) used to demonstrate (a) end-to-end SDN control and (b) delivery of services on a *per subscriber basis* at the same time. As we will present, in our approach there are two different domains where SDN control is applied. One is internal to the NFVI and is based on the Contrail System. The other one spans end-to-end and is based on the OpenDaylight control. Details of the NFVI and control are described in Section 6.1.

Wireless domain: The wireless network of the architecture, consists of the wireless access network and a wired openflow-capable backhaul packet-core network. In more detail, a multi-SSID capable 802.11n testbed is provided [31], together with an LTE network. The LTE network utilizes virtual Access Point Names (vAPNs) for traffic differentiation per MOVNO (together with OVS/Openflow support in the EPC) [37]. Using Multi-SSID technique we are able to create multiple virtual-Access Points (v-APs) that emulate the operations of a physical AP at the MAC level. All these functionalities are exposed to the IML, with the services we describe in Section 4.2. A summary of the hardware specification of the wireless network is provided in Table 1.

Essentially, every 802.11 Access Point (node) is connected to the Openflow packet-core network, while also openflow control is performed in the EPC of the LTE. The path that the egress traffic follows is: user → Access Point (eNB or Wi-Fi) → OpenFlow control points → testbed gateway systems → Geant network → TSON optical → Datacenter and the reverse order for the ingress traffic. The openflow network is used to control different traffic flows on per MOVNO basis. Virtual switches and the functionality provided by the IML and the VICL provide for this flexibility and the provisioning and configuration of the end-to-end paths. In the case where the NFV testbed is also utilized, the traffic before exiting the wireless domain, through tunnels, passed from the NFV testbed where user specific service chains were executed. The concept, design considerations and system requirements alongside use cases that illustrate the advantages of adopting network service chains are presented in [29].

Table 1
Wireless resources specification.

Wi-Fi Net: 120 NITOS nodes, Gigabit eth,wireless interfaces, i7-2600, 8M Cache, 3.40 GHz CPU SSD, 4G HYPERX BLU DDR3, NITlab CM card *Atheros*802.11$a/b/g/n$(*MIMO*), 350 Watt mini-ATX, Multi-band 5dbi,2.4Ghz, 5Ghz, pigtails (UFL to RP-SMA)
LTE Net: EPC SIRANN,ENBs Ip.Access LTE 245F, 3GPP R. 8.9.0, Single Carrier, Dual Band 1/13, 4/13,2/5 or 7/13. 10MHz Band.,2x2 MIMO-Single User DL, S1-IP RF 2 x 10dBm, 16QAM U/L and D/Lm, Max Thr. 13Mbps, # active users: 4, # idle users: 64
UE: USB dongles (e.g., Huawei E392, Huawei E398) OpenFlow: HP 3800 switches(v1.3),Pronto 3290 (v1.0)

Optical domain In the optical segment, TSON technology [46,57] is used as the enabling technology at the physical layer. TSON offers multi-facaded resource virtualization, based on FPGA technology, with guaranteed bandwidth shares per MOVNO by scheduling both wavelengths and time-slots. The TSON dataplane functions can be split in to two types of TSON edge and TSON core functions. TSON edge nodes use FPGA platforms to process and aggregate ingress Ethernet data for generating bursts and transmission over WDM channels. The controller for transmission of optical time-slices programs the nodes. Regarding the TSON core nodes, optical modules are deployed and allow fast optical switching of the bursts of data. The information for switching of optical time-slices is programmable and is information that can be configured by the controller. We highlight that TSON provides bandwidth granularity as fine as 30$Mb/s/\lambda$ stretching up to 9.1$Gb/s/\lambda$ and supports multiple wavelength operation. A summary of the hardware specification of the optical network is provided in Table 2.

GEANT interconnect: The physical inter-connection between the wireless domain (NITOS in Volos, Greece) and the optical testbed (TSON in Bristol, UK) exploited connectivity through the GEANT network [1] (1 Gbps link, vlans 692 and 800). GEANT is the pan-European data network for the research and education community, that interconnects all EU universities and research institutes.

Note that no SDN control could be applied over the GEANT network. However GEANT was the only available choice to interconnect the two remote testbeds; the other alternative would be VPN

Table 2
Optical resources specification.

TSON Dataplane: PLZT fast switches (10 ns), FPGA
Xilinx V6 HTG, passive Mux/Demux for light filtering.
Amplfiers for compensating for switching losses.
shorter bursts of data which carry smaller payloads.
supporting as fine as 30 Mb/s per time slices.
Fix number of time slices in each frame.
Up to 31 separate flows could be aggregated to each frame.

TSON control plane: Server with raw Ethernet
interfaces to FPGAs. Java based agent with REST API.

over the public internet. Although SDN control could not be exploited in that segment, the GEANT network was already evaluated in the context of other experimental projects (EU FED4FIRE, EU FIBRE projects etc.) and it is highly stable. In addition, in order to have at least two real end-to-end paths, two different VLANs were used end-to-end. Note, that this case is as similar to real scenarios, where the ISPs network is interposed between the datacenter and the RAN. The behavior of this segment of the network depends on SLAs with the ISP and no external control can be applied. The vertical interfaces designed and implemented in our work can be exploited in any type of wireless-optical integrated solution, where multi-domain administration restricts some areas where SDN mechanisms can be applied in the end-to-end path. In addition the layering design proposed is flexible to additionally consider for a SDN solution in the wired interconnect, since all the control is made from an end-to-end orchestrator.

SDN/NFV Testbed by Juniper Networks: A Juniper NFV testbed is also integrated to the mutli-domain testbed, consisting of a Juniper MX480 router, based Service Control Gateway function and a Juniper Contrail cloud management platform, hosting vSRX virtual firewall service as a Virtual Network Function (VNF). More details on the SRX, Contrail technology and the Service Gateway used can be found in [30] and the official Juniper web site. The interconnected infrastructure between NITOS and Juniper EMEA SLAB has been implemented using IPSEC VPNs over Internet, terminated at SRX240 firewalls on both locations. Two separated IPSEC tunnels and virtual routers were used on the SRXs to carry and isolate the non-services applied and services applied traffic. The TSON optical testbed provided an Internet exit point for the wireless clients and hosted vSRX VNF and x86 server functions (see Fig. 4 for the network connectivity diagram).

The Service Control Gateway application resides on an MX Series router and is *a subscriber-aware and application-aware* service delivery network edge element, located between the gateway of the access network and the public network and network services. The service control gateway *enforces policy rules based on subscriber-awareness, application-awareness, and service data flow identification*. The Juniper testbed integration gave us the ability to evaluate the multi-domain approach not only as a multi-domain solution but also as a framework that can be used to support NFV concepts.

Thus for the end-to-end service (e.g. a web request for a public web site) with ODL we affect the network elements in the wireless domain and the optical domain, but in the internet and data center Contrail control was in effect. This was also the first implementation that this type of SDN control was demonstrated. Note that in both the domains a MOVNO identifier (actually the vlan id) was used to differentiate services between multiple MOVNOs.

4.2. Design and implementation of the IML

The main functionalities of the IML are related to the virtual (and the physical) resource abstraction, reservation, management and control. It is the element of the architecture responsible for the creation of isolated virtual infrastructures. These are composed of resources from both the network domains, in a uniform way in order to expose common APIs and utilize a common resource lifecycle management mechanism, independently of the type of the resource. The IML is responsible to hide the complexities and API internals from the VICL. In Fig. 5 a high-level architecture of the Infrastructure Management Layer (IML) and the interaction with domain specific services are presented.

The IML implementation is based on the Open Network as a Service (OpenNaaS) OpenNaaS framework (under LGPLv3 license) [8]. It runs inside an OSGI container called Apache Karaf. In OpenNaaS resources are created by providing a resource descriptor to OpenNaaS Resource Manager, which delegates descriptor interpretation to a Resource Repository. Resource Repository's responsibility is to create resources given a descriptor and manage resource life-cycle. A Resource Bootstrapper is responsible of instantiating the resource model, while a capability ActionSet defines which actions should be provided by a driver. OpenNaaS exposes services to the VICL using a REST API.

In order to expose the necessary functionalities to the IML a set of *control and management services* are provided from both the wireless and optical testbeds. In the wireless domain, the core services utilized are the *NITOS Broker* and the *NITOS Manager*.

- **Broker**: is the component that is responsible for the physical resource advertising and the resource reservation, while it controls the slicing of the resources and guarantees isolation. It directly interacts with the IML in order to build the resource descriptors. In more detail, it keeps an inventory with information regarding the available resources and their virtualization capabilities, exposed through a REST or/and a SFA interface [54]. In addition, it exposes a Slice Isolation and Control Service (SICS) to the *Manager* services to infer if the requested actions to the physical resources should be authorized or not. Using the Broker services any type of resource (switch, access point, virtual access point, LTE networks etc.) are exposed to IML.
- ***Manager:*** exposes a secured REST interface to the IML and is responsible for the management, configuration and operation of the physical network. The *Manager* operations rely on ssh-based access to the nodes or the OMF Aggregate Manager [54] and remote commands execution. We note that for the case of Openflow control there was no translation in a REST interface, rather IML was acting like proxy, forwarding the control channel to the OpenFlow switches. The switches state however was exposed in the IML like the rest infrastructure.

A similar approach was taken also in the Optical Domain, where custom XML-RPC services where used for the resource reservation and physical network provisioning. Using the Manager services any type of resource switch, access point, virtual access point, LTE networks etc.) are configured and managed through APIs by the IML.

4.3. Design and implementation of the VICL-SDN control

The VICL functions for the on-demand provisioning of interdomain network connectivity allow to establish dynamic user-to-DC connections from the mobile devices to the VMs placed in the different datacenters, crossing the wireless backhaul and the TSON metro domains. These functions are invoked at the cloud service runtime, when the user wants to access the virtual cloud infrastructure deployed and provisioned via OpenStack, as requested through the Heat interface [33].

The ONF architecture [16] defines two main interfaces, the Application-Controller Plane Interface (A-CPI) and the Data-Controller Plane Interface (D-CPI). In our architecture, the D-CPI corresponds to the interface between the IML and the VICL and provides all the functionalities related to discovery, configuration

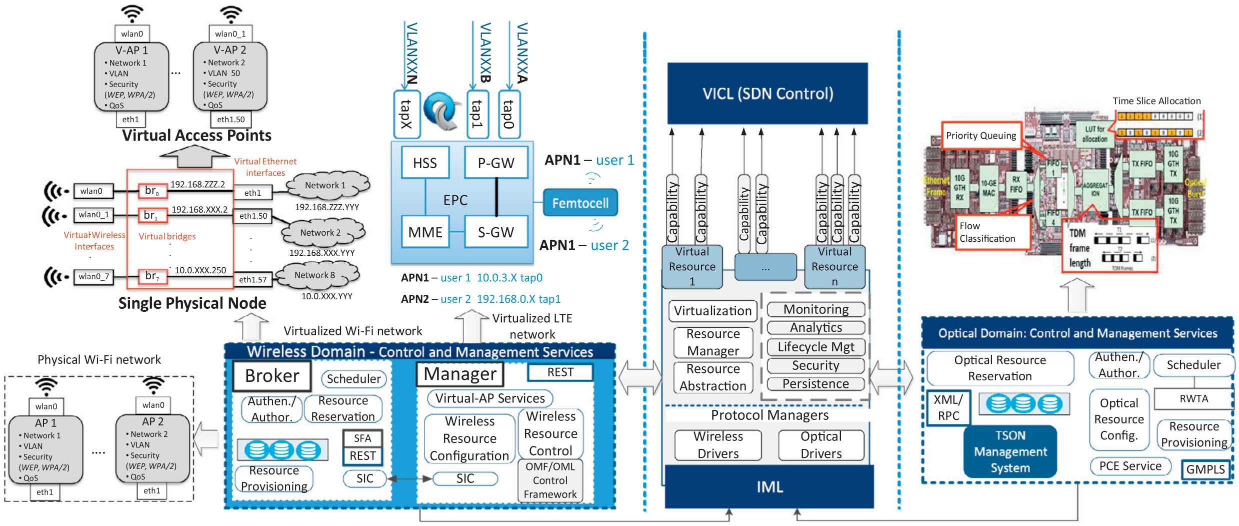


Fig. 5. High-level architecture of the Infrastructure Management Layer (IML) and the interaction with domain specific services. In the wireless domain the virtualization services are provided by the NITOS Broker (for physical and virtual resource reservation) and the NITOS Manager (for the control and management of the physical and virtual resources). The NITOS Broker relies on the SFA API that are now exposed using a REST interface. The NITOS Manager is used to expose the necessary control and management functions for both the virtualized Wi-Fi and LTE networks. For the optical domain a similar API exposure was adopted. Inside the IML a set of functions per MOVNO are exposed like virtual network exposure, resource management, analytics etc. The SDN control is applied on the virtualized network exposed by the IML.

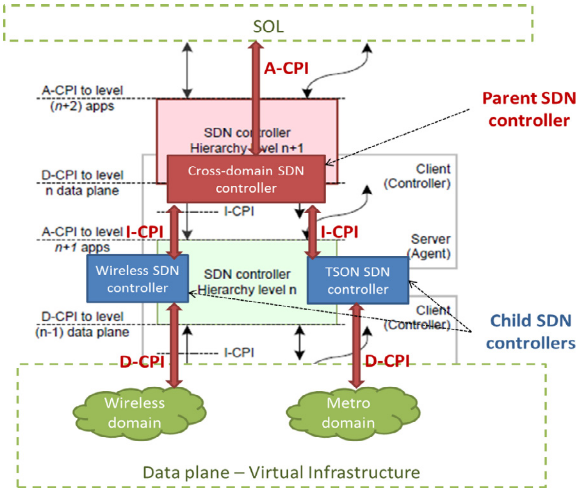


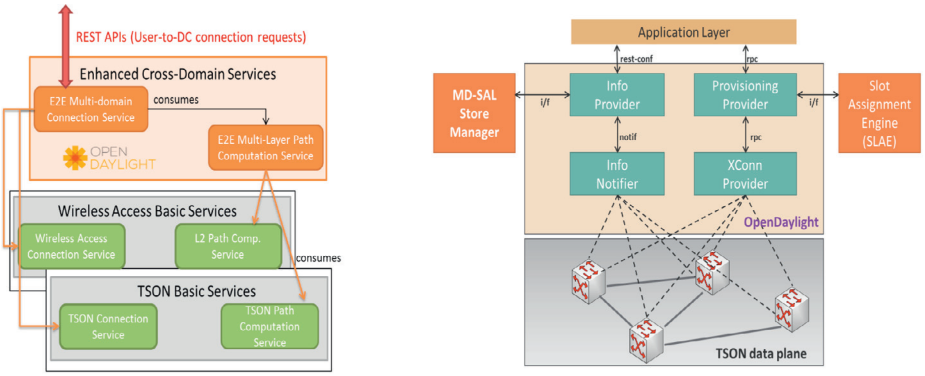
Fig. 6. Mapping of VICL distributed deployment model with the recursive hierarchical distribution of SDN controllers defined by ONF.

and monitoring of the virtual resources belonging to the virtual infrastructure under control of a given SDN controller (of the lower layer) deployed in the VICL. Fig. 6 presents the mapping of VICL distributed deployment model with the recursive hierarchical distribution of SDN controllers defined by ONF. The A-CPI in our case can be considered at two different levels. The former is the interface between the SDN controller providing the elementary per-domain functionalities and the cross-domain applications running on top of that for the operation of the whole infrastructure. A second level of A-CPI is the interface between the entire VICL and the Service Orchestration Layer (SOL), responsible for the management of the end-to-end cloud services. It should be noted that, from the SOL perspective, the actual deployment of the VICL is completely transparent and the VICL itself can be considered as a single SDN controller that exposes some services on its A-CPI to control the end-to-end multi-domain connectivity.

In our solution the VICL is implemented as a logically centralized SDN controller, based on the OpenDaylight Helium version. Note that other integrated SDN solutions like ONOS, can provide a similar set of functionaries required by the VICL. The SDN controller is enhanced with additional plugins (i.e. TSON, wireless and multi-domain managers) which implement all the functions to configure network resources and establish connectivity in all the network domains. Moreover, existing OpenDaylight components have been enhanced to integrate TSON and the wireless resources in OpenDaylight core services (e.g. topology manager) and graphical interface (i.e. dlux component). The specific features we developed are the *odl-features-wireless-provider, Odl-features-tson-provider* and *Odl-features-multidomain-provider*. These implement the procedures to provide network connectivity across TSON domains, OpenFlow-based wireless backhaul networks and end-to-end multi-domain connections respectively.

The end-to-end Multi-Layer Path Computation Service acts as a parent PCE and computes the end-to-end path from a source edge node in the wireless backhaul to a destination edge node in the TSON network which interconnects a datacenter. This computation is performed combining the intra-domain paths elaborated by the Path Computation Services for the TSON and the wireless backhaul domains, implemented as basic functions in the SDN controller. Since all these software components are developed as OpenDaylight modules, their interaction is based on the direct invocation of the methods provided by the java classes auto-generated from the YANG modules of each service. YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol (see RFC 6020 for details). The End-to-End Multi-Domain Connection Service, on the other hand, offers a RESTCONF interface to external components for requesting the setup and tear-down of on-demand paths. This module is a consumer of the End-to-End Multi-Layer Path Computation Service to determine the inter-domain abstract path, composed of edge nodes only.

The actual configuration of the network resources in wireless backhaul and TSON segments are delegated to the Wireless Access Connection Service and the TSON Connection Service for the specification of the API. Fig. 7(a) presents the end-to-end network connectivity services in the VICL, while Fig. 7(b) gives an overview of the TSON software architecture implemented in ODL. The control-plane is a set of applications that exchange data to provide the


REST APIs (User-to-DC connection requests)
Enhanced Cross-Domain Services
E2E Multi-domain Connection Service
consumes
E2E Multi-Layer Path Computation Service
OPEN DAYLIGHT
Wireless Access Basic Services
Wireless Access Connection Service
L2 Path Comp. Service
consumes
TSON Basic Services
TSON Connection Service
TSON Path Computation Service
Application Layer
rest-conf
rpc
MD-SAL Store Manager
i/f
Info Provider
Provisioning Provider
i/f
Slot Assignment Engine (SLAE)
notif
rpc
Info Notifier
XConn Provider
OpenDaylight
TSON data plane

(a) The VICL: End-to-end Network connectivity services.

(b) TSON support and components in ODL

Fig. 7. VICL implementation based on the OpenDaylight controller.

full management of the data-plane, i.e. the ODL controller, the Sub Lambda Allocation Engine (SLAE) and the Application layer.

- *Info Notifier:* it establishes and maintains the connections with the physical devices (TSON data-plane) and provides a description of the capabilities in terms of equipment, ports and resources.
- *XConn Provider:* it sends the provisioning commands to the devices.
- *Info Provider:* it stores the notification information and discovers the underlying topology.
- *Provisioning Provider:* it manages the incoming RPCs provisioning messages and computes the best path in the network (through the SLAE-Manager).
- *Sub-wavelength Lambda Allocation Engine (SLAE):* it computes the best path into the TSON data-plane.
- *MD-SAL In-Memory Store Manager:* it stores information related to nodes, node-connectors and links providing the discovered topology to the other modules of the ODL controller.

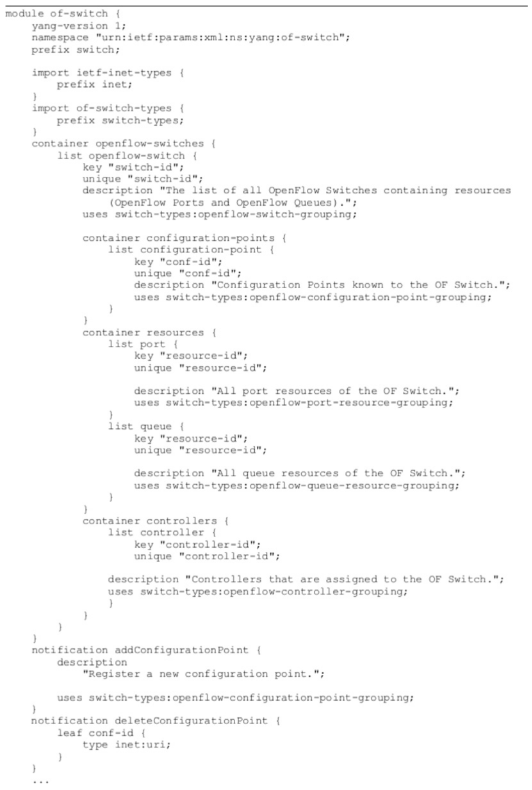
A similar approach was followed for the wireless domain, with *XConn Provider* etc., for the provisioning of the network connectivity on the openflow-based backhaul network. In Fig. 8 part of the YANG definition of the of-switch in CONTENT is presented.

4.4. Design and implementation of the Service Orchestration Layer

The Service Orchestration Layer (SOL) is based on an extended version of the OpenStack cloud management platform, with some software components enhanced to add the capability to interconnect different data centers with a given QoS in terms of guaranteed bandwidth per MOVNO.

OpenStack has become the de facto cloud control and management framework. It is actually a set of software tools for building and managing cloud computing platforms for public and private clouds.

In our approach OpenStack Neutron and OpenStack Heat have been extended with the definition of a new resource, which represents a QoS-enabled connection between two different zones. In the integrated testbed, Neutron interacts with the OpenDaylight controller (implementing the VICL functions), using the ML2 (Modular Layer) plugin with a configuration that enables the OpenDaylight driver. Both ML2 plugin and OpenDaylight driver have been properly modified to support the new network resources. The OpenStack Juno version was used, where the deployment model is based on a simple architecture that includes a server where both OpenStack controller node and network node are installed and two compute nodes that use KVM as hypervisor. The controller and network node runs the Identity Service (Keystone), the Image Service (Glance), the management modules of Compute and Network Services (Nova and Neutron), the Neutron ML2 plugin and several agents to provide network services and external connectivity, as well as the Orchestration Service (Heat). The compute nodes run the hypervisor part of the Compute Service (Nova) that operates the tenant virtual machines (VMs), the Neutron ML2 plugin and an agent to interconnect tenant networks to virtual machines.



```
module of-switch {
    yang-version 1;
    namespace "urn:ietf:params:xml:ns:yang:of-switch";
    prefix switch;

    import ietf-inet-types {
        prefix inet;
    }
    import of-switch-types {
        prefix switch-types;
    }
    container openflow-switches {
        list openflow-switch {
            key "switch-id";
            unique "switch-id";
            description "The list of all OpenFlow Switches containing resources
                (OpenFlow Ports and OpenFlow Queues).";
            uses switch-types:openflow-switch-grouping;

            container configuration-points {
                list configuration-point {
                    key "conf-id";
                    unique "conf-id";
                    description "Configuration Points known to the OF Switch.";
                    uses switch-types:openflow-configuration-point-grouping;
                }
            }
            container resources {
                list port {
                    key "resource-id";
                    unique "resource-id";

                    description "All port resources of the OF Switch.";
                    uses switch-types:openflow-port-resource-grouping;
                }
                list queue {
                    key "resource-id";
                    unique "resource-id";

                    description "All queue resources of the OF Switch.";
                    uses switch-types:openflow-queue-resource-grouping;
                }
            }
            container controllers {
                list controller {
                    key "controller-id";
                    unique "controller-id";

                description "Controllers that are assigned to the OF Switch.";
                uses switch-types:openflow-controller-grouping;
                }
            }
        }
    }
    notification addConfigurationPoint {
        description
            "Register a new configuration point.";

        uses switch-types:openflow-configuration-point-grouping;
    }
    notification deleteConfigurationPoint {
        leaf conf-id {
            type inet:uri;
        }
    }
    ...
```

Fig. 8. YANG file definition for the wireless domain.

Technical Approach: Fig. 9 shows the steps required for the deployment and provisioning of a distributed cloud service, where VMs can be potentially located in different data-centers. Authentication and authorization actions are not shown for simplicity, but are handled at the OpenStack level through the Keystone module. For simplicity we focus on the optical segment and a similar approach is considered for the dynamic configuration of the wireless

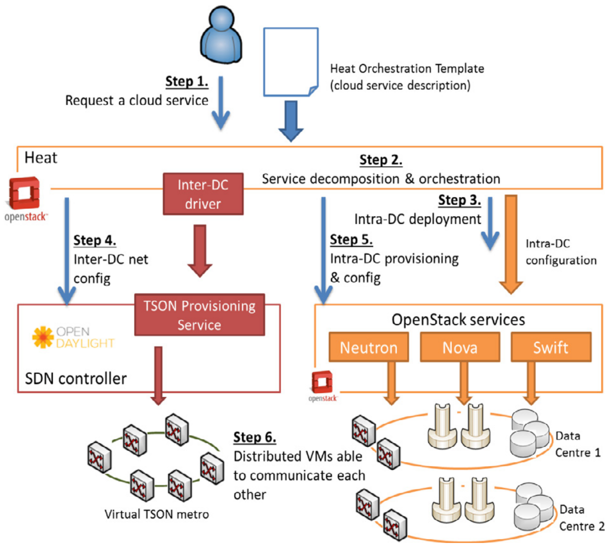


Fig. 9. Workflow for the provisioning of a cloud service.

backhaul. In general, we are assuming that a standard, best-effort connectivity to allow the exchange of control messages (i.e. to allow the user to connect to OpenStack API from the mobile device) is pre-configured and always maintained.

As depicted in Fig. 9 an external user (i.e. MOVNO) can request an end-to-end connectivity with a guaranteed bandwidth value, which is mapped over an end-to-end path crossing the wireless backhaul and the TSON metro network, up to the datacentre. The mechanisms to establish in a dynamic manner the user-to-DC connectivity are handled by the SDN controller, as described in the previous section, and are invoked through the REST API of an End-to-End Multi-Domain Connection Service. Depending on the scenario, user-to-DC connectivity can be manually requested and pre-configured through management actions or dynamically adapted to the mobile flows. The second option requires an automated detection and recognition of the traffic flows for the different subscribers and the different cloud services. Finally, at the edges of the network, traffic must be properly tagged on a per-service basis to guarantee that each flow is directed through the proper path in the entire user-to-DC segment. Considering the capabilities of the TSON edge nodes, which discriminate the ingress traffic depending on VLAN IDs or destination MAC address, in CONTENT we adopt the VLAN tagging. This choice imposes a limitation on the maximum number of services which can be supported.

Cloud Use cases: Network QoS in distributed cloud computing services is fundamental for applications like Content Delivery Networks, which require the transfer of a large amount of data with bandwidth, delay and jitter constraints to guarantee a suitable Quality of Experience for the end-users. On the other hand, QoS for the inter-DC connections is also a requirement to handle cloud management traffic (e.g. for VMs migration or backups) in an efficient manner, without any negative impact on running cloud applications.

OpenStack-OpenDaylight integration: The interaction between OpenStack and OpenDaylight exploits the SDN controllers REST APIs. In this scenario, OpenDaylight offers APIs to establish inter-DC network connections, classifying the traffic based on VLAN IDs. The OpenStack network module (i.e. Neutron) is extended with a new network resource that defines the interconnectivity between VMs and the related QoS parameters. As part of the VMs creation workflow, Neutron interacts with OpenDaylight acting as a client and requests the creation of network connections between the hosts where the VMs are placed, specifying the parameters for QoS (minimum requested bandwidth, maximum delay) and traffic classification (VLAN IDs selected by OpenStack for the virtual networks where the VMs are attached).

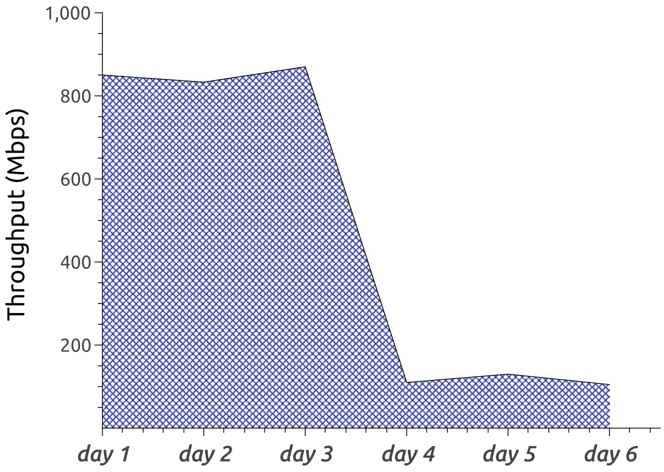


Fig. 10. Average UDP performance for 5 days samples, 10 samples per day.

5. Performance evaluation

5.1. Evaluation of the Physical Infrastructure Layer

The integrated system, the proposed architecture, the vertical interfaces built and their implementation between the Physical underlay, the Virtual Infrastructure Layer and the SDN control plane, are among the first to be presented and evaluated in a realistic end-to-end fashion. However note that in production systems, the wireless domain network and the optical domain network must be physically collocated, in order to achieve maximum throughput performance.

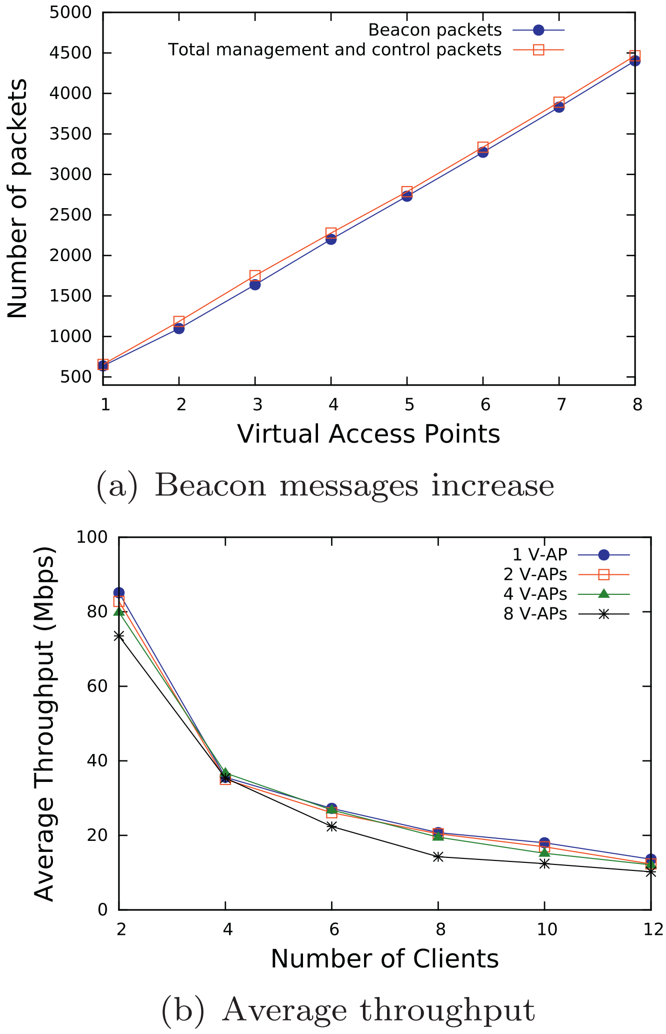
Link Characterization The performance over the end-to-end link was quite volatile in terms of bandwidth and latency. The main reason was that the NITOS testbed endpoints carry real shared traffic for numerous services, before exiting to the GEANT network, thus affecting end-to-end communication (see Fig. 10 for the average UDP performance using 6 days samples, 10 samples per day.

As reference values, averaged for the first three days period, we report 8.86 Mb/s downlink and 7.85 Mb/s uplink for TCP, 800 Mb/s downlink and 626 Mb/s uplink for UDP, latency 64.2 ms, jitter 0.072ms. On average, TCP performance was always worse than UDP. As we present also in Section 6 a number of reasons affect these values. The idea is that using the SDN approach, through feedback we can stabilize performance on per MOVNO and per flow basis on given values. Initial results presenting the wireless-optical data-plane integration and data-path performance were reported in [44].

5.2. Evaluation of the IML

Using the testbed specific control & management services, all the physical and virtualized resources from both the testbeds were exposed to the IML. Within the IML a common resource abstraction representation mechanism (using the OpenNaaS system) facilitated the provisioning of the Virtual Infrastructures.

Fig. 11 presents the effects of using virtual Multi-SSID based Access Points (APs), instead of physical ones. Although virtualization of the AP, is desired in multi-operator environments, in the Multi-SSID approach comes with a signaling overhead increase, as presented in Fig. 11(a), for a varying number of V-APs (1 V-AP, 2-VAPs, 4V-APs and 8V-APs). The beacon messages increase leads to an



(a) Beacon messages increase

(b) Average throughput

Fig. 11. Multi-SSID virtualization cost.

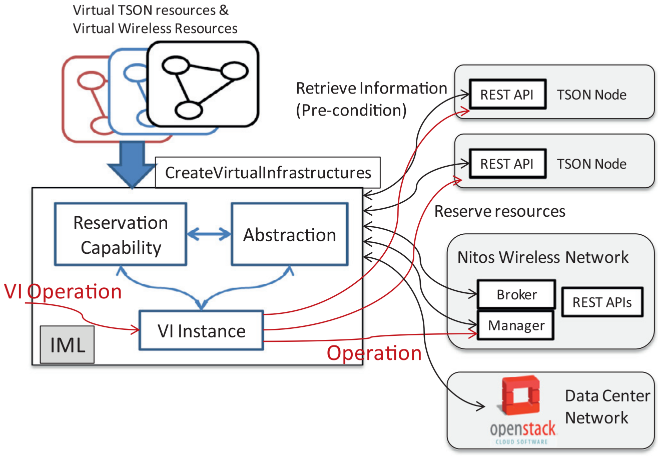


Fig. 12. Composition of virtual resources on top of physical infrastructures.

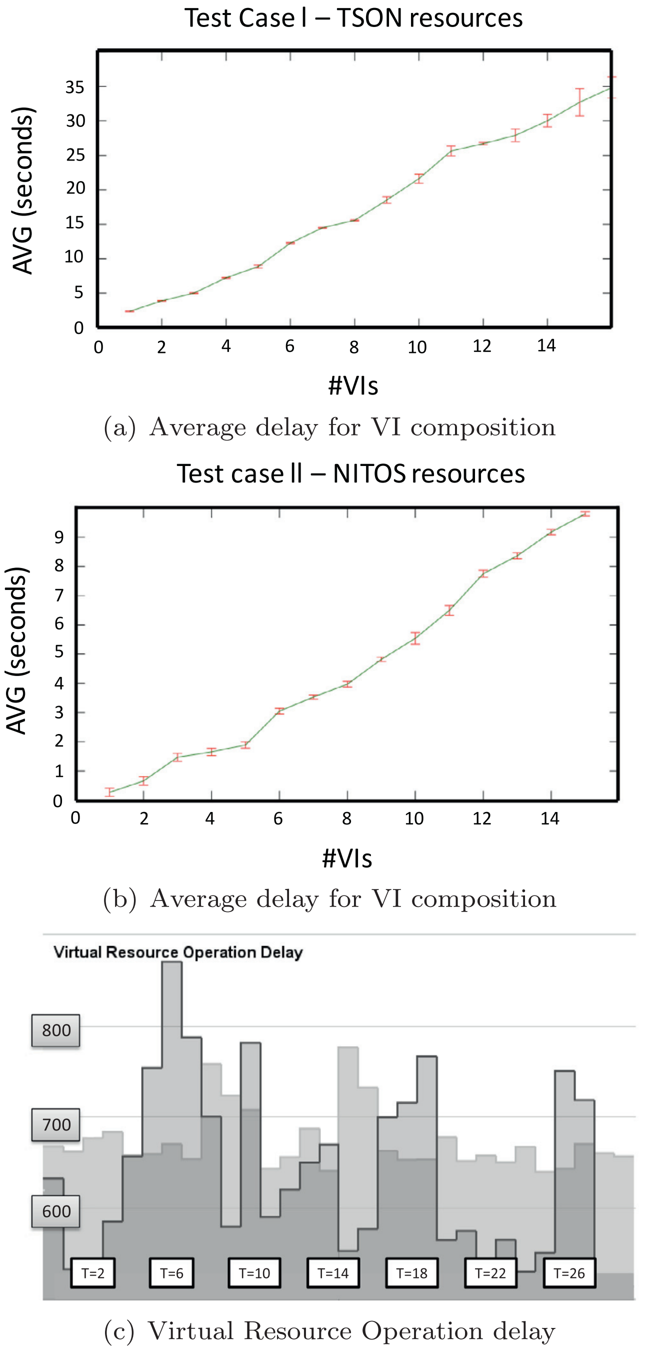
overall decrease of the average throughput (Fig. 11(b)), since more bandwidth is utilized for signaling packets. See [31] for a detailed analysis of the Multi-SSID approach.

In order to evaluate the IML performance, while provisioning virtual network infrastructures (VI), we have executed a number of sub-test cases: VI composed of optical resources, VI composed of wireless resources, VIs composed of both optical and wireless resources and VIs composed of optical and wireless resources coupled with IT resources. See Fig. 12 for a visual representation of the execution scheme. We highlight that for the TSON nodes, we perform resource-based virtualization, meaning that all the virtualization logic resides and it is executed within the IML, while for the wireless we perform service-based virtualization, meaning that all the virtualization logic resides outside the IML and it is only invoked from the IML. More details on the service versus resource virtualization models can be found in [14].

Virtual Optical Resources: The test generates a set of VI requests. Each request contains three virtual TSON nodes connected between them, and for each virtual link we generate a random amount of time slots required (e.g. between one-time slot and ten times slots). For each iteration, we add one virtual infrastructure request, which means that the first iteration requests one VI, the second contains two VI requests, and so on. We iterate from 1 to 15 VI requests, and execute each test case for 30 repetitions, so we ensure that the results are valid. Fig. 13(a) depicts the results obtained after the execution of this test case. The initial analysis clearly depicts a reasonable tendency to increase the VI provisioning time as the load of requests increases: the higher the number of simultaneous VI requests the IML receives, the longer it takes to provision these. Considering the size of the PHY substrate and the amount of simultaneous VIs requests, the IML still performs under reasonable terms, e.g. 20 s for provisioning 10 simultaneous VIs. This means that in the case where each VI is composed of three resources, and there are 10 simultaneous VIs, the IML system is capable of provisioning, in around 20 s all the VIs and all the virtual resources. In terms of traditional planning times, with non-virtualized networks, this process may take several hours or even days.

Virtual Wireless Resources: This test generates a set of VI requests, where each request contains one virtual wireless node (e.g. a wireless 802.11n node), since granularity at service-level is set to the level of a whole node. For each iteration of the test, we add one virtual infrastructure request, which means that the first iteration requests one VI, the second contains two VI requests, and so on. Again, we iterate from 1 to 15 VI requests, and execute each test case for 30 repetitions, so we ensure statistical validity of the results. Fig. 13(b) depicts the results obtained after the execution of the whole test case. The analysis clearly depicts a reasonable tendency to increase the VI provisioning time as long as the load of requests increases, similarly to the previous case of the Virtual Optical Resources. The higher the number of simultaneous VI requests the IML receives, the longer it takes to provision these. Nevertheless, considering the size of the PHY substrate and the amount of simultaneous VIs sent, the virtualization here takes place faster than in the case of the optical resources. Note that the NITOS Broker performs the reservation of resources only, instead of planning all the requests. For the operation of the virtual resources the *NITOS Manager* services are called by the IML, where depending on the action requested different response times are experienced (spanning from ms time scale for simple network configuration actions that can reach up to 80 s for actions like loading the Operating System through PXE services). In terms of time scales, it is still reasonable to generate VIs within the seconds scale, in comparison to the amount of work required in actual deployments without considering virtualization.

Integrated framework performance: In order to ensure that the inclusion of the whole virtual layer does not affect too much the performance, we have run a set of tests in order to identify the delay introduced by the IML in the virtual infrastructure operation. Since all the IML offers REST APIs on the northbound side, we have run the experiments for both GET and POST operations.



(a) Average delay for VI composition

(b) Average delay for VI composition

(c) Virtual Resource Operation delay

Fig. 13. Evaluation results regarding the composition of virtual resources on top of physical infrastructures in the IML.

Fig. 13(c) depicts the different executions for each operation (horizontal axis), and the delay measured in milliseconds (vertical axis). GET operations are depicted in light grey, while POST operations are depicted in dark grey. The results demonstrate that the average delay for GET and POST operations is 673,138ms and 635,505ms respectively. Standard deviation is around 33 and 90 ms respectively. Overall, the results are reasonable and as expected from the performance perspective, considering the gains directly derived from infrastructure sharing. Regarding data center operations the IML relies on a specific Cloud Management System (CMS)(i.e. OpenStack) in order to manage the reservation and provisioning of virtual machines, without taking it into account during the planning phase. Since the CMS is only contacted during the planning phase, the results for end-end setup can be directly derived from the previous test cases, as it considers exactly the same procedures (Fig. 14).

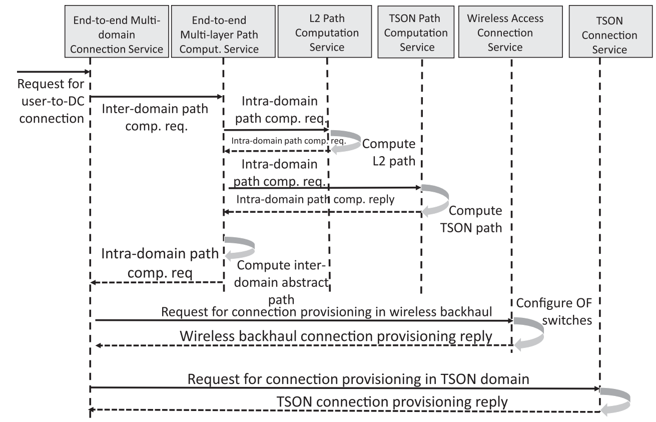

End-to-end Multi-domain Connection Service
End-to-end Multi-layer Path Comput. Service
L2 Path Computation Service
TSON Path Computation Service
Wireless Access Connection Service
TSON Connection Service
Request for user-to-DC connection
Inter-domain path comp. req.
Intra-domain path comp. req
Intra-domain path comp. req.
Compute L2 path
Intra-domain path comp. req.
Intra-domain path comp. reply
Compute TSON path
Intra-domain path comp. req
Compute inter-domain abstract path
Request for connection provisioning in wireless backhaul
Configure OF switches
Wireless backhaul connection provisioning reply
Request for connection provisioning in TSON domain
TSON connection provisioning reply

Fig. 14. Workflow for provisioning of user-to-DC connectivity.

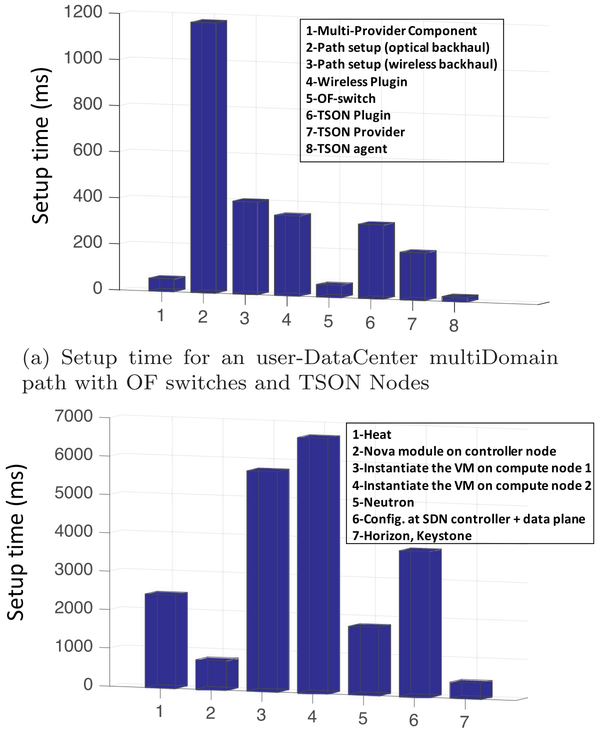
5.3. Evaluation of the VICL

A set of test cases was used to demonstrate the capability of the SDN control components developed to provide QoS-guaranteed network connectivity for inter Data Center (inter-DC) (communication across DCs) and user-to-DC communications over TSON and wireless domains. The objective was to verify the VICL procedures to establish a multi-domain path across the wireless backhaul and the optical metro domains of the integrated testbed. This path is used to provide a mobile user with QoS guaranteed access to cloud services, in our case to a VM located in the data center.

The main entity involved is the SDN controller and involves the modules related to the TSON technology and the components dedicated to the provisioning of QoS-enabled connectivity in the OF-based wireless backhaul. See Fig. 7(a) for the software components involved. The test cases execution is triggered directly from the OpenDaylight DLUX interface, which has been enriched to allow the administrator to setup a multi-domain path and guide him/her in the configuration of all the required parameters, like source and destination end points and resources.

The delay of translating all the operational commands and queries from the virtual to the physical resources is introduced by the IML, in order to implement multi-tenant virtualization and has been measured as follows (specific to our setup). For GET operations the average time of 673.138 ms, with 33 ms standard deviation and for POST operations average time of 635.505 ms, with 90 ms standard deviation. For scenarios where the SDN controller and cloud orchestrator are managed by the MOVNOs over virtual infrastructures, the service provisioning time must be modified to take into account the additional delay introduced by the IML.

- *End-to-end setup for an inter-domain path.* The end-to-end setup for an inter-domain path that we demonstrate involved a single OpenFlow switch in the wireless backhaul and three TSON nodes and has an average time of 1631 ms. The contribution of each component on the end-to-end setup time is presented in Fig. 15(a). In all cases average values are reported. For example the processing time at the multidomain-provider component equals 56 ms, the time required for path setup in the wireless backhaul 402 ms and the time required for path setup in the TSON domain 1173 ms. The result of the procedure to process a new request is a message at the north-bound RESTCONF API of the multi-domain provisioning application developed.



(a) Setup time for an user-DataCenter multiDomain path with OF switches and TSON Nodes

(b) Setup time for an user-DataCenter multiDomain path with OF switches and TSON Nodes

Fig. 15. Evaluation results regarding end-to-end connectivity in CONTENT.

5.4. Evaluation of the Service Orchestration Layer

In this test case we measured the time required for the on-demand setup of a cloud service. The cloud service is composed by VMs placed on two different computed nodes in the data center, interconnected through the TSON network. The entire test case is triggered from the user through the OpenStack dashboard, providing a Heat template, which describes the desired virtual environment. The request is processed by the orchestrator component (Heat), which decomposes the overall service specification in its elements, i.e. VMs, networks, subnets, etc.). These include the new resources defined to describe an inter-DC connection. Then, the orchestrator interacts with the different clients to deploy each specific type of resource inside the data center (the intra-DC segment) for VMs and intra-DC network, and in the segment that involves additional the optical network (the inter-DC segment) for the configuration of the TSON metro network. The first action is internal in OpenStack and involves the interaction between Heat and Nova or Neutron, while the second action is carried out through an interaction between Heat and OpenDaylight. Finally, the virtual resources that have been deployed in the previous step are activated and configured and the VMs are able to communicate with each other. The whole setup time for cloud service takes an average time of 21622 ms where in Fig. 15(b), we present the average processing time values. At the cloud orchestrator level, as expected, the major delay is introduced by the time required to instantiate the VMs on the two compute nodes (a total of 12406 ms), compared to a global processing time at the OpenStack controller node of 5415 ms. The high value of the network side configuration (3801 ms compared with the 1150 ms measured in the first test) depends on the additional actions required to configure the OpenvSwitch instances on the two compute nodes. This configuration requires the involvement of the neutron-service and OVSDB plugin at the SDN controller level (legacy components not modified) and the interaction between OpenDaylight and the two compute nodes using the OVSDB and OpenFlow protocols.

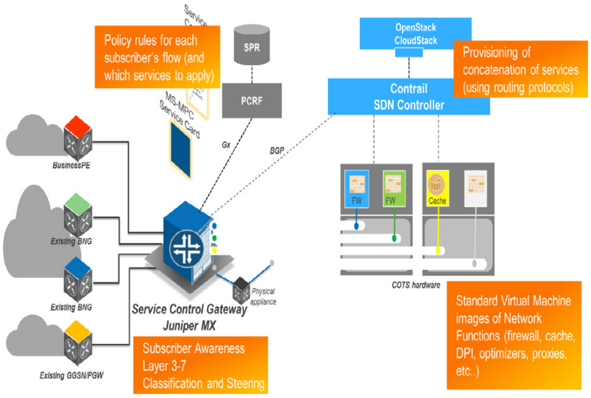

Service
Policy rules for each subscriber's flow (and which services to apply)
SPR
PCRF
MS-MPC Service Card
Gx
BGP
OpenStack CloudStack
Contrail SDN Controller
Provisioning of concatenation of services (using routing protocols)
BusinessPE
Existing BNG
Existing BNG
Existing GGSN/PGW
Service Control Gateway Juniper MX
Physical appliance
Subscriber Awareness Layer 3-7 Classification and Steering
FW
FW
Cache
COTS hardware
Standard Virtual Machine images of Network Functions (firewall, cache, DPI, optimizers, proxies, etc..)

Fig. 16. SCG functional solution.

6. Use case demonstrations

In this section we demonstrate the application of end-to-end SDN control in two use cases. The first is the case where the integrated system was used to support NFV-based parental control. The second is related to the demonstration of video applications and service differentiation in a multi-operator environment.

6.1. Multi-domain, multi-control SDN in support of NFV

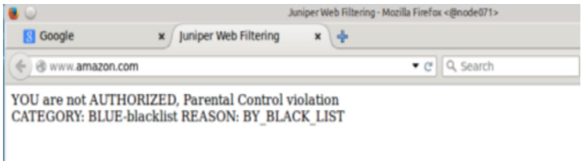
Using the extended testbed, presented in Fig. 4, we demonstrated the use case of a customized web parental control security function (URL filtering/blacklisting) provided by Contrail hosted vSRX (FW-BLUE) instance, leveraging the Unified Threat Management (UTM) functions of vSRX.

SDN/NFV Testbed by Juniper Networks: Services In Fig. 16 a high level representation of the SCG solution is presented. Note that in this domain of the end-to-end architecture a different SDN controller is used to control the infrastructure, namely the Contrail system [3]. Thus at the same time we apply SDN control over the end-to-end data path using the ODL solution and the Contrail/SCG system is used for the necessary traffic steering and Service chaining responsible to deliver per subscriber service provisioning.

Policy control and traffic steering using Juniper's, SCG and Contrail SDN Control: Policy control and traffic steering for mobile traffic are supported through e.g. the use of the JUNIPER Service Control Gateway (SCG). The Juniper's SCG software package can be split into two software building blocks. The Control Plane related software modules that manage the subscriber state machine such as commonly used signaling interfaces and events like accounting and usage analytics records generation.

SCG, Contrail and Service Chaining: Service chains are a concatenation of network functions that can be deployed as physical or virtualized functions. They can be represented as an ordered directed graph of functions that are concatenated (with a provisioning process) to implement specific processing capabilities. The role of the SDN controller is to automate the concatenation of the functions without having to build the concatenation in the underlay network infrastructure itself.

The commonly recommended approach for service chaining automation is to use overlays rather than building the service chain directly on the underlay infrastructure. In principle, controllers may use standard MPLSoGRE / MPLSoUDP / VXLAN tunnels to create the user plane overlay topology. In the case of the Contrail controller, BGP can be used to propagate routing information to extend a service chain to physical router performing the steering



YOU are not AUTHORIZED, Parental Control violation
CATEGORY: BLUE-blacklist REASON: BY_BLACK_LIST

Fig. 17. Parental control violation example using per-subscriber service chaining in CONTENT.

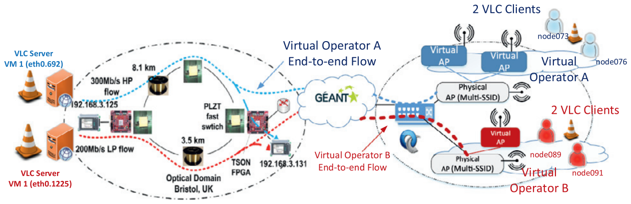
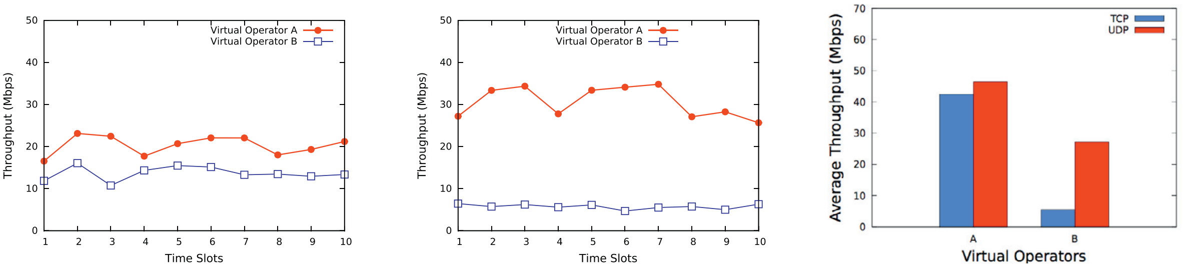


Fig. 18. Two MOVNO providers with end-to-end flows, from the data center up to the wireless user. Every MOVNO has two users that are served by virtualized APs.

of per subscriber flows. Finally the controller communicates with XMPP to a virtual Router (vRouter), a software element that under the control of the SDN controller, running in the x86 hypervisor - builds a service chain topology in the virtualized environment.

Use case execution: The customer traffic was redirected by configured policy on the MX Service Control Gateway to and from the vSRX security function. The test involved a wireless client simulating the subscriber device initiating web browsing requests towards different websites. Based on the configured parental control policy requests to sites categorized as online shopping or gambling resulted in a block action with a response message (see Fig. 17). Browsing to other non-blacklisted websites was allowed. Service Control Gateway can use PCC and ePCC rules to redirect, or steer, a subscribers traffic to a third-party server to apply a service chain see. The rule identifies a routing instance to apply to the traffic (Fig. 18).

To stress this fact, we also executed another use cases relevant with FTP transfers. In this example we configured a bandwidth rate limit per subscriber, for uplink and downlink traffic classified as a FTP transaction. FTP traffic matching the configured DPI rules on SCG will be rate limited based on the configured maximum bit rate (MBR) settings (256Kbps download/128Kbps upload) FTP server running on UBris premises, hosting 10MB test files for download. For the case of 10MB uplink the transaction varied from 673.75 s (15.20 KB/s) to 334.83 s (30.58 KB/s). On one hand there was the rate limit, on the other hand there was great variability on the network conditions). Nevertheless by means of feasibility study the system was functional and the limit was applied when necessary.



(a) Without Service Differentiation

(b) With Service Differentiation on per MOVNO basis (VLAN)

(c) With Service Differentiation on per MOVNO basis (TCP-UDP)

By means of performance, we report that in the case the chain was already instantiated and under no congestion the control was applied in 450ms on average, while in order to instantiate service instances in Contrail for the first time, the time scale varies in values above 10secs.

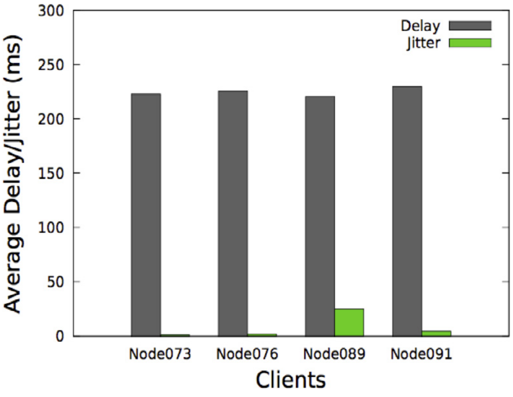


Fig. 19. With service differentiation on per MOVNO basis (Delay-Jitter).

Table 3
Differentiation profiles.

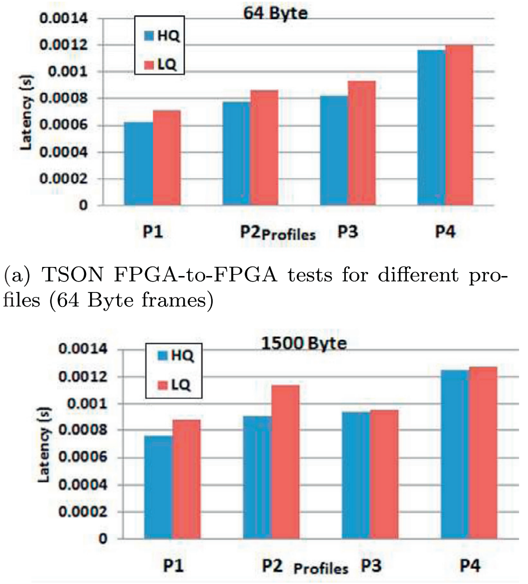
	Frame allocation	ms
P1	Non contiguous(100100..)	0.716
P2	Contiguous(11000)	0.716
P3	Non contiguous (100100..)	1.331
P4	Contiguous (11000)	1.331

6.1.1. Multi-operator environment, end-to-end service differentiation

Regarding the network evaluation of the end-to-end path, we performed various experiments in order to extract sufficient measurements and present data plane and control plane efficiency. From this extended set of experiments we preset a VLC streaming application, with Wi-Fi networks, connected through the TSON network with the datacenter endpoint, as depicted in Fig. 19. In this setup using the SDN approach described we control the end-end-flows of two MOVNO providers, with the control points being in the optical segment and the wireless backhaul. Every MOVNO has two users that are served by virtualized Multi-SSID-based APs.

In one hand we used end-to-end SDN for provisioning the end-to-end paths with differentiating services for two competing MOVNOs. We also used our integrated SDN interfaces to change priorities in real time. All the APs operate in the 2.4GHz band, while VMs in the Bristol side where configured as the VLC servers.

We identified flow 1 as a flow that carries high priority traffic and we allocated a 300Mb/s channel end-to-end, from the servers that reside on the data center, up to the wireless APs. Similarly, we identified flow 2 as low priority and we allocated a 200Mb/s



(a) TSON FPGA-to-FPGA tests for different profiles (64 Byte frames)

(b) TSON FPGA-to-FPGA tests for different profiles - 1500 bytes frames

Fig. 20. High encoded rate video samples.

channel end-to-end. Note that the behavior of the network in the GEANT network was not controlled using the SDN framework.

In the TSON Traffic differentiation methods were applied to the flows using exemplar allocation (1 means allocated) and frame durations as shown in profiles P1-P4 in Table 3. The profiles P1-P4 show combinations of frame lengths and allocation patterns (contiguous or not by the scheduler) which impact the latency. In TSON we used three sub-wavelength switching TSON nodes (FPGA + PLZT fast switch) in a ring configuration, and fiber spools added on links to emulate transmission distances on different paths.

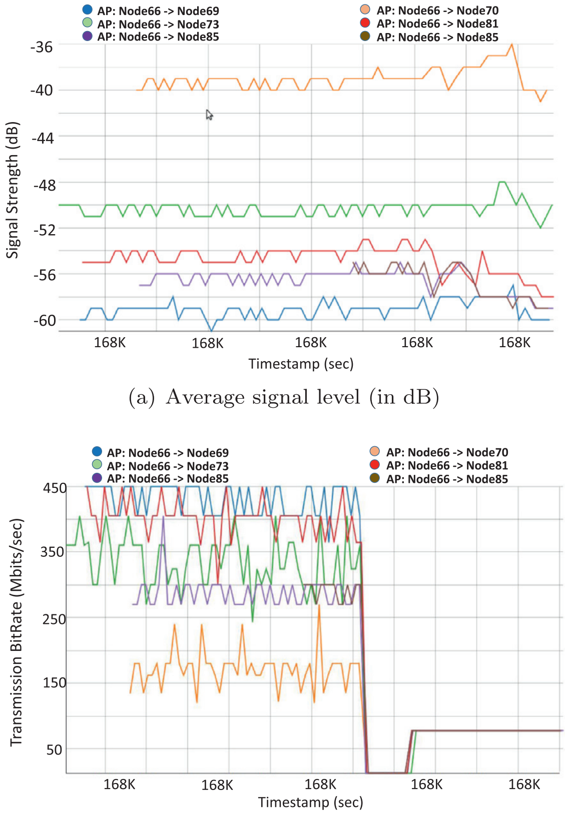
Fig. 20 shows the average throughput for 10 time slots. We made experiments with a high encoded rate blu ray video sample (duration 147 s - 373.1 MB). In Fig. 19(a) no service differentiation was performed, while in Fig. 19(b) we applied QoS based policies in the openflow switches, served as the backhaul for the APs, based on VLAN identification and VLAN priorities. Fig. 19(b) presents both the TCP-UDP performance, where we highlight that in all cases and during all the experiments contacted TCP traffic always had worst performance. Fig. 21(b) presents the average end-to-end delay/jitter performance.

For the same experiment, Figs. 20(a) and (b) show the results for 64 and 1500 Byte Ethernet frame sizes for FPGA-to-FPGA communications in TSON respectively. It is observed that the shorter TSON TDM frame, using high priority queues, and with more distributed allocation of time-slices leads to faster traffic delivery. A similar testbed where the TSON technology is exploited is described in [25].

7. Lessons learned and future directions in multidomain SDN

The main components that were introduced in the CONTENT solution and are novel in existing state of the art are the following:

- First time real-life demonstration of a multi SDN control environment, where one domain was controlled by ODL and the other from Contrail SDN control systems.
- It was the first time ODL plugins were developed for the programmability of the optical TSON network. These developments were later exploited/enhanced in the context of 5G-XHAUL solution.



(a) Average signal level (in dB)

(b) Changing the transmission rate in the APs

Fig. 21. Demonstration of factors affect final performance.

- Demonstration of orchestration over openstack in multi-domain environments in pan-european level.
- Virtualization of FIRE testbed, using flexible REST interfaces that are SFA aligned and built over the OMF control framewrok. Note that OMF is also the control framework used in GENI testbeds in US (like Orbit).
- Demonstration for per MOVNO service offering and control. These types of services are gaining paramount importance in the context of 5G Network slicing.
- Demonstration of multi-domain QoS provisioning and difficulty of maintaining the QoS end-to-end. The DiffServ model operation and multiple QoS points greatly affect the final performance. Although SDN can provide the necessary tools to adjust dynamically the performance at each control point, sophisticated feedback-based algorithms still needs to be developed.

At this point and as a concluding remark, we highlight a number of challenges we had to consider and that are still open towards true wireless-optical integration and application of the SDN paradigm in production systems.

Too many parameters affect final performance: end-to-end performance is affected by a large number of factors, since multiple control points exist in the end-to-end path and many stochastic in nature communication channels. An indicative list of such factors is the following:

1. Number of users: The number of connected users, significantly affects the throughput every MOVNO will achieve on average. For example in the case of 802.11 networks, due to the CSMA-CA method, each client will share the total throughput of the channel in a proportional way, depending on factors such as distance of the AP, signal level (RSSI), etc.

2. *Wireless Channel Quality:* Dense deployments, Channel Quality, interference and a number of stochastics are factors that affect the final performance. For example in Fig. 21(a) we present the average signal level for a cluster of 802.11 APs we used for our experiments. While this information can be exposed to the SDN controller and the policies in effect, the extreme variability in time is something still difficult to handle.
3. *RAN configuration choices:* Either we are discussing about LTE eNBs or WiFi APs, depending on the base station configuration, we can affect the total rate provided to all the associated clients and limit it to some preferable threshold. For e.g. the default rate for an AP is managed by the Minstrell algorithm. In Fig. 21(b) we change the operation of the AP in runtime and set the transmission rate to a preferable value. We exposed a *Manager* service (wrapping *fixed rate idx* of the ath9k driver) that can be called by the SDN controller or even the IML. If in the end-to-end path a phenomenon like this is observed is extremely difficult to identify the reason of this performance degradation.
4. *Openflow SDN Rules in the switch fabric:* Using the Openflow switch we can differentiate the physical rate of each flow providing different throughput to the end-to-end users. In more recent designs Openflow is also used to affect performance in the EPC [38,48]. *Service Differentiation in TSON:* Using the scheduling capabilities that the TSON technology offers, we can provide to every MOVNO specific bandwidth guarantees and effectively differentiate services. Without careful design these actions can be contradictory.
5. Geant/ISP interconnects: Although, this is not the case in C-RAN designs, but it is for other types of muti-domain networking, in our experimental setup we also need to consider for the GEANT network performance. In other practical deployments ISP policies also greatly affect the end-to-end path.

Extremely difficult troubleshooting: From implementation point of view, this is one challenge we had to meet and we believe is extremely important towards true multi-domain SDN. Inherent LTE complexities, SDN southbound and northbound interfaces design and operation, network connectivity problems, bad configuration choices are all candidates as sources of failure. Although the network is much more efficient,it is also much more complex with a set of SDN/NFV functions and services offered. Still, the procedures to troubleshoot still rely on traditional “naive” ways. We believe this is an open field where a radically different approach is required in order to facilitate network services and functions troubleshooting.

Multi-domain orchestration: Following the work delivered under the concept of SDN/NFV is not only the ability to apply SDN principles at domain specific segments of the network. It is also the multi-domain orchestration that is required for true convergence and network integration. Indeed, there is still a lot of work to be done and many challenges to be met towards 5G communications, in order to satisfy the extreme requirements set by means of performance. Towards this direction, the proposed solution provided some preliminary results,presenting that this convergence is feasible even in the orchestration layer of end-to-end multi-domain schemes.

Relation with NFV and alignment with ETSI MANO: With the proliferation of cloud-based technologies and towards integrated 5G communications, numerous architectures have been already proposed. As we described in the related work section, such efforts are presented for example in [5,19,36] with respect to the SDN/NFV and the cloud computing design paradigms. In our solution a Juniper specific design was adopted for the NFVI while also for the relevant NFV management and orchestration procedures. Like also in all the architectures proposed, also in our case the relation of the ETSI MANO [20] core components with the SDN/NFV frameworks proposed, is an open question. Multi-domain orchestration procedures are also in the core of very active research and is part of future planning.

The purpose of this paper is to present a feasibility study and analyze the shortcomings and challenges when it comes in the applicability of end-to-end SDN solutions. The relevant optimizations especially through advanced orchestration procedures are a very hot research topic and currently an open field. The novelty of our approach resides on the architecture, the interfaces definition and implementation. These can actually offer the necessary tools that intelligent algorithms running on the orchestration level can be used to optimize the network.

8. Conclusions

In this work we presented the design and evaluation results for the open, multi-domain SDN system called CONTENT, while focusing on the necessary abstractions and virtualization techniques to integrate virtual wireless and optical resources. We demonstrated how state of the art SDN control mechanisms and cloud management frameworks were extended in order to support an end-to-end network virtualization system. We also presented how an end-to-end system can be used to support NVF concepts. Evaluation results were provided and implementation experience was reported using a wireless-optical integrated testbed. We demonstrated how network programmability using the SDN approach promotes agility and automation of the procedures, while at the same time brings that network provisioning times and configuration to orders of magnitude less than the traditional approaches.

Acknowledgments

This work has been supported by the EU Project n. 318514 “Convergence of wireless Optical Network and IT rEsources iN support of cloud services” (CONTENT). We would like to thank G. Grammel from JUNIPER networks for all his valuable help and material for NFV testbed provided by JUNIPER networks. We would also like to thank Roberto Monno from Nextworks and Adrian Rosello from I2CAT for all their help in carrying out the experiments.

References

[1] Geant network, (http://www.geant.org).
[2] NITOS wireless testbed, (http://nitlab.inf.uth.gr/NITlab/).
[3] Contrail Architecture. White paper, Juniper Networks, 2013.
[4] M. Agiwal, A. Roy, N. Saxena, Next generation 5g wireless networks: a comprehensive survey (2016).
[5] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, A. Benjebbour, Design considerations for a 5g network architecture, Commun. Mag. IEEE 52 (11) (2014) 65–75.
[6] I.F. Akyildiz, P. Wang, S.-C. Lin, Softair: a software defined networking architecture for 5g wireless systems, Comput. Netw. 85 (2015) 1–18.
[7] H. Ali-Ahmad, C. Cicconetti, A. La Oliva, M. Draxler, R. Gupta, V. Mancuso, L. Roullet, V. Sciancalepore, Crowd: an sdn approach for densenets, in: Software Defined Networks (EWSDN), 2013 Second European Workshop on, IEEE, 2013, pp. 25–31.
[8] I. Bueno, J.I. Aznar, E. Escalona, J. Ferrer, J. Antoni Garcia-Espin, An opennaas based sdn framework for dynamic qos control, in: SDN4FNS, IEEE, 2013, pp. 1–7.
[9] J. Carapinha, J. Jiménez, Network virtualization: a view from the bottom, in: Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, ACM, 2009, pp. 73–80.
[10] P. Chanclou, A. Pizzinat, F.L. Clech, T.L. Reedeker, Y. Lagadec, F. Saliou, B.L. Guyader, L. Guillo, Q. Deniel, S. Gosselin, S.D. Le, T. Diallo, R. Brenot, F. Lelarge, L. Marazzi, P. Parolari, M. Martinelli, S. O'Dull, S.A. Gebrewold, D. Hillerkuss, J. Leuthold, G. Gavioli, P. Galli, Optical fiber solution for mobile fronthaul to achieve cloud radio access network, in: 2013 Future Network Mobile Summit, 2013, pp. 1–11.
[11] C.Y. Chang, N. Nikaein, T. Spyropoulos, Impact of packetization and scheduling on c-ran fronthaul performance, in: IEEE GLOBECOM, 2016, pp. 1–7.
[12] M. Channegowda, R. Nejabati, M.R. Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, et al., Experimental demonstration of an openflow based software-defined optical network employing packet,

fixed and flexible dwdm grid technologies on an international multi-domain testbed, Opt. Express 21 (5) (2013) 5487–5498.

[13] A. Checko, H.L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M.S. Berger, L. Dittmann, Cloud ran for mobile networksa technology overview, Commun.Surveys Tutorials 17 (1) (2015) 405–426.

[14] N.M.K. Chowdhury, R. Boutaba, Network virtualization: state of the art and research challenges, Commun. Mag. IEEE 47 (7) (2009) 20–26.

[15] N.M.K. Chowdhury, R. Boutaba, A survey of network virtualization, Comput. Netw. 54 (5) (2010) 862–876.

[16] O.M.E. Committee, et al., Software-Defined Networking: The New Norm for Networks, ONF, 2012. White Paper

[17] J. Costa-Requena, Sdn integration in lte mobile backhaul networks, in: Information Networking (ICOIN), 2014 International Conference on, IEEE, 2014, pp. 264–269.

[18] S. Das, G. Parulkar, N. McKeown, P. Singh, D. Getachew, L. Ong, Packet and circuit network convergence with openflow, in: Optical Fiber Communication Conference, Optical Society of America, 2010, p. OTuG1.

[19] A. De La Oliva, X.C. Pérez, A. Azcorra, A. Di Giglio, F. Cavaliere, D. Tiegelbekkers, J. Lessmann, T. Haustein, A. Mourad, P. Iovanna, Xhaul: toward an integrated fronthaul/backhaul architecture in 5g networks, Wireless Commun. 22 (5) (2015) 32–40.

[20] G. ETSI, 001 (mano), online https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.

[21] M. Fiorani, S. Tombaz, J. Martensson, B. Skubic, L. Wosinska, P. Monti, Modeling energy performance of c-ran with optical transport in 5g network scenarios, IEEE/OSA J. Opt. Commun. Netw. 8 (11) (2016) B21–B34.

[22] N.J. Gomes, P. Chanclou, P. Turnbull, A. Magee, V. Jungnickel, Fronthaul evolution: from cpri to ethernet, Opt. Fiber Technol. 26 (2015) 50–58.

[23] S. Gringeri, N. Bitar, T.J. Xia, Extending software defined network principles to include optical transport, Commun. Mag. IEEE 51 (3) (2013) 32–40.

[24] A. Gudipati, D. Perry, L.E. Li, S. Katti, Softran: software defined radio access network, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ACM, 2013, pp. 25–30.

[25] J. Gutiérrez, N. Maletic, D. Camps-Mur, E. García, I. Berberana, M. Anastasopoulos, A. Tzanakaki, V. Kalokidou, P. Flegkas, D. Syrivelis, et al., 5G-xhaul: a converged optical and wireless solution for 5g transport networks, Trans. Emerging Telecommun.Technol. 27 (9) (2016) 1187–1195.

[26] G. Hampel, M. Steiner, T. Bu, Applying software-defined networking to the telecom domain, in: INFOCOM, 2013 Proceedings IEEE, 2013, pp. 3339–3344.

[27] G. Hampel, M. Steiner, T. Bu, Applying software-defined networking to the telecom domain, in: Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, IEEE, 2013, pp. 133–138.

[28] I. Hwang, B. Song, S.S. Soliman, A holistic view on hyper-dense heterogeneous and small cell networks, Commun. Mag. IEEE 51 (6) (2013) 20–27.

[29] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu, Research directions in network service chaining, in: SDN4FNS, IEEE, 2013, pp. 1–7.

[30] Juniper Networks, The NFV Service Edge, 2015. White paper. https://www.juniper.net/assets/cn/zh/local/pdf/solutionbriefs/3510520-en.pdf.

[31] K. Kousias, K. Katsalis, D. Stavropoulos, T. Korakis, L. Tassiulas, Building virtual 802.11 testbeds towards open 5G experimentation, WCNC, IEEE, 2015.

[32] D. Kreutz, F.M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.

[33] R. Kumar, N. Gupta, S. Charu, K. Jain, S.K. Jangir, Open source solution for cloud computing platform using openstack, Int. J. Comput. Sci.Mobile Comput. 3 (5) (2014) 89–98.

[34] C. Liang, F.R. Yu, Wireless network virtualization: a survey, some research issues and challenges, Commun. Surv. Tutorials IEEE 17 (1) (2015) 358–380.

[35] C. Liu, K. Sundaresan, M. Jiang, S. Rangarajan, G.-K. Chang, The case for re-configurable backhaul in cloud-ran based small cell networks, in: INFOCOM, 2013 Proceedings IEEE, IEEE, 2013, pp. 1124–1132.

[36] J. Liu, T. Zhao, S. Zhou, Y. Cheng, Z. Niu, Concert: a cloud-based architecture for next-generation cellular systems, Wireless Commun. IEEE 21 (6) (2014) 14–22.

[37] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, L. Tassiulas, Enabling open access to lte network components; the nitos testbed paradigm, in: NetSoft, IEEE, 2015, pp. 1–6.

[38] S. Matsushima, R. Wakikawa, Stateless user-plane architecture for virtualized epc (vepc), draft-matsushima-stateless-uplane-vepc-04 (work in progress) (2015).

[39] V.-G. Nguyen, T.-X. Do, Y. Kim, Sdn and virtualization-based lte mobile network architectures: a comprehensive survey, Wireless Pers. Commun. 86 (3) (2016) 1401–1438.

[40] B.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: past, present, and future of programmable networks, Commun. Surv. Tutorials IEEE 16 (3) (2014) 1617–1634.

[41] A. Panda, W. Zheng, X. Hu, A. Krishnamurthy, S. Shenker, Scl: Simplifying distributed sdn control planes, USENIX NSDI, USENIX Association, 2017.

[42] A. Patro, S. Banerjee, Coap: a software-defined approach for home wlan management through an open api, ACM SIGMOBILE Mobile Comput. Commun.Rev. 18 (3) (2015) 32–40.

[43] J. Qadir, N. Ahmed, N. Ahad, Building programmable wireless networks: an architectural survey, EURASIP J. Wireless Commun. Netw. 2014 (1) (2014) 1–31.

[44] B. Rahimzadeh Rofoee, K. Katsalis, et al., First demonstration of service-differentiated converged optical sub-wavelength and lte/wifi networks over geant, in: Optical Fiber Communication Conference, Optical Society of America, 2015, p. Th2A.35.

[45] M. Rahman, C. Despins, S. Affes, Hetnet cloud: leveraging sdn & cloud computing for wireless access virtualization, in: Ubiquitous Wireless Broadband (ICUWB), 2015 IEEE International Conference on, IEEE, 2015, pp. 1–5.

[46] B.R. Rofoee, G. Zervas, Y. Yan, D. Simeonidou, Griffin: programmable optical datacenter with sdn enabled function planning and virtualisation, J. Lightwave Technol. 33 (24) (2015) 5164–5177, doi:10.1109/JLT.2015.2440480.

[47] V. Sagar, R. Chandramouli, K. Subbalakshmi, Software defined access for hetnets, Commun. Mag. IEEE 54 (1) (2016) 84–89.

[48] M.R. Sama, S. Ben Hadj Said, K. Guillouard, L. Suciu, Enabling network programmability in lte/epc architecture using openflow, in: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on, IEEE, 2014, pp. 389–396.

[49] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, L. Mathy, Network virtualization architecture: proposal and initial prototype, in: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures, ACM, 2009, pp. 63–72.

[50] R. Trivisonno, R. Guerzoni, I. Vaishnavi, D. Soldani, Sdn-based 5g mobile networks: architecture, functions, procedures and backward compatibility, Trans. Emerging Telecommun.Technol. 26 (1) (2015) 82–92.

[51] A. Tzanakaki, M.P. Anastasopoulos, S. Peng, B. Rofoee, Y. Yan, D. Simeonidou, G. Landi, G. Bernini, N. Ciulli, J.F. Riera, E. Escalona, J.A. Garcia-Espin, K. Katsalis, T. Korakis, A converged network architecture for energy efficient mobile cloud computing, in: International Conference on Optical Network Design and Modeling, 2014, pp. 120–125.

[52] D.P. Venmani, O. Le Moult, F. Deletre, Y. Lagadec, Y. Morlon, On the role of network synchronization for future cellular networks: an operator's perspective, IEEE Commun. Mag. 54 (9) (2016) 58–64.

[53] R. Vilalta, V. López, A. Mayoral, et al., The need for a control orchestration protocol in research projects on optical networking, in: EuCNC, IEEE, 2015, pp. 340–344.

[54] T. Wauters, B. Vermeulen, et al., Federation of internet experimentation facilities: architecture and implementation, in: European Conference on Networks and Communications (EuCNC 2014), 2014, pp. 1–5.

[55] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, L. Zeng, Openran: a software-defined ran architecture via virtualization, in: ACM SIGCOMM Computer Communication Review, 43, ACM, 2013, pp. 549–550.

[56] S. Yeganeh, A. Tootoonchian, Y. Ganjali, On scalability of software-defined networking, IEEE Commun. Mag. 2 (51) (2013) 136–141.

[57] G.S. Zervas, J. Triay, et al., Time shared optical network (tson): a novel metro architecture for flexible multi-granular services, Opt. Express 19 (26) (2011) B509–B514.

[58] S. Zhu, J. Bi, C. Sun, C. Wu, H. Hu, Sdpa: Enhancing stateful forwarding for software-defined networking, in: ICNP, IEEE, 2015, pp. 323–333.

K. Katsalis (male) is a post-doctoral researcher in the Mobile Communications Department at Eurecom, France (http://www.eurecom.fr). His research interests fall in the area of Mobile communications, SDN/NFV and cloud computing technologies, network and systems virtualization, Wireless communications, Stochastic Control Theory, Distributed Systems, QoS, Modeling and management of computer networks.

Kostas Katsalis was a Research partner in the Informatics and Telematics Institute (ITI) (http://www.iti.gr) of the Center for Research and Technology Hellas (CERTH-ITI) between 2010 and 2015. He holds a Msc (2007) and a PhD degree (2015) from the Electrical and Computer Engineering Department from the University of Thessaly, Greece (http://www.inf.uth.gr) under the supervision of professor Leandros Tassiulas. He has large experience in EU Funded projects (FP7 CONTENT, RECITAL, FLEX, H2020 XHAUL, COHERENT), while he also has two years industry experience (2008–2010) in Mobile Communications and services, InternetQ, Athens, Greece (http://www.internetq.com).