End-to-end Network Slicing for 5G in Multi-Region, Multi-Tenant Cloud Platform

Simona Marinova, Thomas Lin, Hadi Bannazadeh, and Alberto Leon-Garcia

Dept. of Electrical and Computer Engineering, University of Toronto
Toronto, ON, M5S 3G4, Canada
{simona.marinova,t.lin,hadi.bannazadeh,alberto.leongarcia}@utoronto.ca

***Abstract*— End-to-end network slicing represents an auspicious concept, which promises to lead the way towards achieving the 5G service requirements. Based on network softwarization and virtualization, it is capable of enabling network as a service (NaaS) for different vertical industries and allowing operators to deploy multiple virtual networks on shared physical infrastructure. In this paper, we implement a network slicing framework, and elaborate on the building blocks. We start by summarizing the key technologies that enable the realization of network slices, then we give an overview of our implementation and show how it provides the tools to support network slicing. Additionally, we provide an initial evaluation to validate the proposed slicing model.**

Keywords— End-to-end (E2E) network slicing, Software Defined Networking, Network Function Virtualization, Management and Orchestration, 5G.

I. Introduction

The next generation mobile network *(5G)* represents a complete transformation rather than an improvement upon the previous systems. The defined use cases and applications that it will support can fit into three main service types [1]: *Enhanced mobile broadband (eMBB)* for bandwidth-hungry applications; *Massive machine type communications (mMTC)* for high-volume, dense IoT applications; and *Ultra-reliable and low-latency communications* for mission-critical services. 5G will enable coexistence of human-centric and machine type communications, which will naturally lead to a large diversity of communication characteristics.

In order to deploy the different 5G services in an economically efficient way, namely capital and operational expenditures, the various services need to share the physical infrastructure. Furthermore, operators need to be able to dynamically manage the lifecycle of the services (i.e., instantiation, resource allocation, healing, decommissioning, etc.), something that conventional architectures cannot support. Hence, 5G is comprehensively searching for architectural solutions and technology pillars to address the resource sharing, management and orchestration.

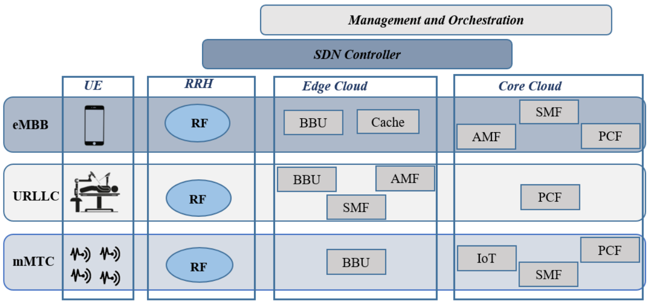
Evolving network softwarization technologies such as *network function virtualization (NFV)* [2], and *software-defined networking (SDN)* [3] laid the foundation to create a flexible architecture, capable of fulfilling a range of service requirements. NFV enables network programmability by decoupling network functions (NFs) from the underlying hardware, whereas SDN decouples the network data and control planes and enables their independent development.

End-to-end (E2E) network slicing is a paradigm built upon SDN and NFV, which emerges as a 5G enabler by systematically addressing the diversified requirements and traffic characteristics of new services, leading to new key performance indicators (KPIs) and quality of service (QoS) metrics. A network slice (NS) represents a logical network that provides specific network capabilities and features with logical isolation [4]. Thus, E2E network slicing puts emphasis on the architectural design changes that are related to the cloud-native nature of 5G in order to create flexible, demand-oriented and agile 5G system, able to address the customer requests in a timely and cost-efficient manner. Fig. 1 depicts services that have different characteristics and require different configuration (i.e., NF placement schemas and resource allocation); however, they have to coexist on the same physical infrastructure where their operation is supervised by a management and orchestration entity.

The advantages of discarding the one-size-fits-all model and designing each NS based on the user demands, are twofold:

Dynamic resource allocation. NFV has provided the ability to implement the NFs on general purpose hardware (GPP). Consequently, computing resources for the specific virtualized network functions (VNFs) can be easily allocated and dynamically changed based on resource utilization and demand in real-time. The aim is to achieve optimal resource allocation: satisfy the user demands, while minimizing the amount of hardware resources and operational costs.

Slice customization. Slices can be dynamically created and configured. Thus, in an SDN-enabled network, each slice can be custom-tailored with regards to the quality of service (QoS) parameters. Also, slices can incorporate additional customized NFs, which is possible due to the software implementation.



RRH – Radio Resource Head; RF- Radio Frequency; BBU - Baseband Unit; AMF – Access and Mobility Management Function; SMF – Session Management Function; PCF – Policy Control Function; IoT – Internet of Things.

Fig. 1. Coexistence of the different 5G services.

In this paper, we design and implement an E2E slicing platform. First, we evaluate the NFV Management and Orchestration (MANO) alternatives, and elaborate on the requirements that have to be fulfilled. Afterwards, we pinpoint the most important building blocks and describe in detail how they interconnect. The integration in a multi-region and multi-tenant cloud platform is shown, as well as preliminary evaluations.

TABLE I. OPEN SOURCE MANO SOLUTIONS

	ETSI OSM	ONAP	OPENBATON	OPENSTACK TACKER	CLOUDIFY
MATURITY	High	Medium	Medium	High	High
SUPPORTED VIMS	OpenStack, OpenVIM, AWS, VMware	OpenStack Will be extended in future.	Openstack Can be extended using plugins.	OpenStack	OpenStack, VMware, Azure. Can be extended using plugins.
SUPPORTED VNFMS	Juju charms	Juju charms	Custom	Generic	Custom
DESCRIPTOR LANGUAGE	YANG YAML	TOSCA YAML	TOSCA YAML	TOSCA YAML	TOSCA YAML
COMMUNITY SUPPORT	High	High	Low	Low	Low
WORKING COMPLEXITY	Medium	High	Medium	Medium	Medium

II. E2E NETWORK SLICING PLATFORM

In this section we elaborate on the E2E slicing framework and briefly review the building blocks.

A. Significance of the NFV Architecture

NFV is a core technology for network slicing, by making the NFs hardware independent, and enabling deployment at any location. Every NS can be granularly decomposed into a set of VNFs, which can be executed on commodity servers. The ETSI NFV [2] model consists of multiple building blocks: NFV Infrastructure (NFVI), the underlying physical infrastructure providing the network, storage and computing resources for the VNFs; VNFs, the main building block, providing the software implementation of the NFs; ETSI NFV MANO framework [5], responsible for management and orchestration of NFV blocks, for both physical and virtual infrastructure.

The ETSI MANO framework consists of different components which are providing the following functionalities: Virtualized infrastructure manager (VIM), entity responsible for the management of the underlying infrastructure; VNF manager (VNFM), responsible for the lifecycle of the VNFs (i.e., creation, scaling, fault management, termination, etc.); NFV Orchestrator (NFVO), responsible for the E2E orchestration of NSs, the constituent VNFs and utilized resources.

B. Management and Orchestration Framework

We have examined 5 different implementations aligned with the ETSI MANO model, and our findings are summarized in Table I. The important characteristics of the open source solutions for our framework were the level of maturity, working complexity, and community support. Based on this, we chose ETSI Open Source MANO (OSM) [6] as the baseline solution for our E2E network slicing platform.

OSM is an ETSI-hosted solution which delivers a functional model of the aforementioned MANO components. Our experience doing installation, customization and evaluation, proved that the level of maturity of the platform is high. Furthermore, it has gained the support from important stakeholders and the open source community, making the development and subsequent releases significantly more stable. OSM supports multiple VIM solutions, including OpenStack, which is the baseline cloud operating system for our platform.

The OSM stack operations can generally be divided in two parts: design and run-time scope. The design-time tools from OSM include role-based authorizations, where users have access to specific projects defined by the administrators. The graphical user interface enables overview of the functionalities which include: catalogs with VNF and NS descriptor files based on YAML configuration files; catalogs with deployed VNFs and NSs, where the actions include creation, deletion, configuration using Juju hooks; and easy integration of new VIMs and SDN controllers. On the other hand, the run-time scope represents a superset of the functionalities defined by ETSI MANO, among which is the automated E2E service orchestration.

Fig. 2 depicts the architectural design of our E2E slicing platform with MANO based on OSM Release 5 [6], which has started supporting network slicing. This release uses the *lightweight lifecycle manager (LCM)*, *resource orchestrator (RO)*, and a *VNF configuration and abstraction (VCA)* module, for the E2E orchestration. RO is the building block responsible for coordinating the configuration and allocation of heterogeneous resources (compute, network, storage), by communicating with the underlying VIMs (the cloud platform northbound interface (NBI), Fig. 2). It also enables the management of various SDN controllers, and writing plugins for custom controllers. The VCA module is the component responsible for the VNFM functionalities defined by ETSI. It supports the initial configuration of the VNFs by

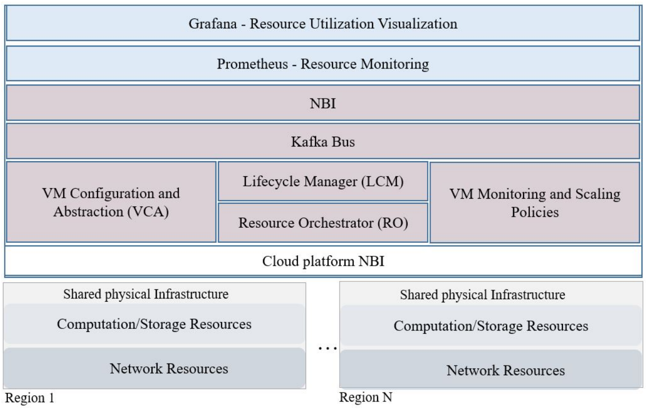

Grafana - Resource Utilization Visualization
Prometheus - Resource Monitoring
NBI
Kafka Bus
VM Configuration and Abstraction (VCA)
Lifecycle Manager (LCM)
Resource Orchestrator (RO)
VM Monitoring and Scaling Policies
Cloud platform NBI
Shared physical Infrastructure
Computation/Storage Resources
Network Resources
Region 1
...
Shared physical Infrastructure
Computation/Storage Resources
Network Resources
Region N

Fig.2. End-to-end Management and Orchestration framework.

using the open source application modelling tool Juju [7]. The main mechanism behind the configuration are the Juju charms which include a selection of software scripts (called hooks) for package installation and YAML files for custom configuration. The LCM module provides the capabilities of the NFVO. It is responsible for the lifecycle management (i.e., creation, deletion) of E2E NSs composed of multiple VNFs by exchanging information and coordinating decisions with the RO and VCA. It is also responsible for the modeling and management of the VNF and NS catalogs, including the descriptors and packaged solutions. It also serves as the interfacing point with external operator blocks, such as operation and business support systems (OSS/BSS). OSM Release 5 introduced enhancements in the *monitoring* and *policy management* modules. In our platform, the monitoring module is used as a tool to drive the data to an external monitoring system. The policy manager is used for creation, management and triggering of alarms based on the infrastructure information (i.e., Gnocchi and Aodh services). The alarms are created as a part of the vertical scaling rules based on metric thresholds, similar to [8].

C. Monitoring and Visualization of Resource Utilization

To acquire more accurate information about the resource utilization from the deployed VNFs, as well as the underlying infrastructure, we have deployed a logically centralized monitoring system and time-series database, *Prometheus* [9]. Prometheus implements a high-dimensional data model, where the time-series are identified by a metric name and a set of key-value pairs. We are looking for the utilization of compute resources such as CPU and RAM, and Prometheus offers querying data in specific time intervals. Furthermore, the configuration file for each monitored node includes an option to customize the scraping period for resource utilization information, depending on the application requirements. Prometheus also offers different exporters for OS metrics, depending on the type of virtualization technology used, hypervisor or container-based, referred to as compute nodes in the following sections. The analytics platform for our solution is *Grafana* [10], a popular tool for querying, visualization and alerting for the utilization of the desired metrics, which allows integration of different databases and monitoring solutions. We integrated and customized a logically centralized solution (i.e., monitoring the overall physical platform, other applications, and compute nodes), in order to obtain resource utilization of the whole heterogeneous infrastructure [11], and be able to perform an informed NS placement.

III. Proof of Concept and Performance Evaluation

This section elaborates on the setup of our E2E design and deployment of a fully virtualized LTE network. First, we describe the cloud platform which facilitates our experiments, and the integration of OSM. Then, we describe the fully virtualized cellular system setup, and the selected open source tools. In the end, we demonstrate two experiments involving all the components of the E2E slicing, and the initial performance evaluations.

A. Logically Centralized OSM on a Multi-Region Platform

Our cloud platform is a **multi-region** and **multi-tenant** testbed [12], which provides distributed NFVIs and leverages SDN for advanced network management [13]. It is based on OpenStack Queens as the VIM [14], a release that is compatible with OSM. This platform is an ideal testbed for our E2E slicing framework, as it can support multi-region and multi-tenant infrastructures in which the different VNFs can be designed and placed depending on the type of service that is being instantiated. In order to keep persistent information about the instantiated network services across multiple tenants, we implemented OSM as a logically centralized solution, Fig. 3. In terms of the scalability, the most important metric is the deployment time as the number of NS increases, which was proven not to depend on the number of consecutive deployments [15].

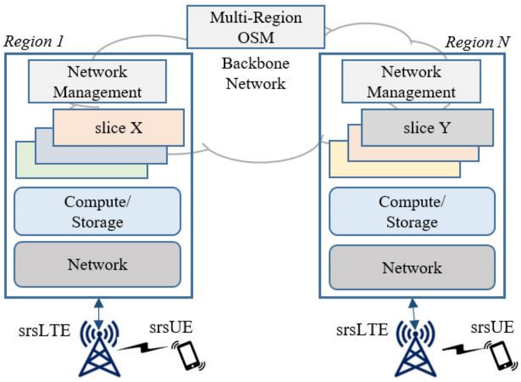


Fig. 3. E2E network slicing platform setup.

B. Virtulized LTE Cellular System Platform

The 5G system is not easily available for experimentation, so we perform the E2E network slicing experiments using a fully virtualized LTE system. Since the framework is based on highly programmable and adaptable technologies, the transition to 5G is expected to be straightforward.

Our virtualized LTE cellular platform implements the Cloud Radio Access Network (C-RAN) [16], featuring centralized processing performed in the cloud, thus facilitating on-the-fly dynamic allocation of the physical hardware to adapt to changing conditions. Hence, the goal is to create a fully virtualized, cloud-based cellular network, through the utilization of open source tools and commercial hardware peripherals.

1) Sofware Defined Radio

The functionalities for the RAN node, i.e., Evolved Node B (eNodeB), are implemented using software-defined radios (SDRs), devices capable of implementing hardware functionalities using software tools. In our cellular platform, we use Universal Software Radio Peripheral (USRP) X310 devices [17], developed by National Instruments. The USRPs are flexible and easily reconfigurable, allowing implementation of radio access technologies through open source software tools which are installed in compute nodes in the cloud. Our E2E slicing framework deploys the LTE system as a NS and monitors its resource utilization, enabling the possibility to dynamically allocate compute resources based on the current network load [18].

2) Virtualized LTE System

We have used *srsLTE* which is an open source library that provides tools for deploying a fully virtualized LTE system (i.e., eNodeB and lightweight Core Network components), as well as LTE user equipment (UE) capabilities [19].

Furthermore, the code base is very modular and intuitive, making it easy to customize and analyze the performance. In terms of hardware, it is executed on compute nodes, and it is compatible with any RF front-end. In order to interface with the X310 USRP devices, srsLTE uses the Ettus Universal Hardware Driver (UHD). The capabilities and performance evaluations of srsLTE have been extensively elaborated in [20], [21]. Table II shows the LTE configuration we have used for our evaluations.

TABLE II. LTE SYSTEM PARAMETERS

System parameters	Parameter configuration
LTE standard	3GPP: Release 8
No. of LTE mobile	1
LTE resource blocks	75
Modulation	64QAM
LTE antenna configuration	MISO
Throughput	Up to 30 Mbps

C. *Performance Evaluation*

Due to paper length constraints, we focus on demonstrating the agility, flexibility and programmability of our slicing framework by conducting two distinct experiments: measuring the E2E NS deployment time, and SDN-enabled traffic shaping. Fig. 3 depicts the proof of concept setup, where the management and orchestration stack is based on OSM Release 5. It is installed in a compute node with 8GB of RAM and 4 CPU cores, in an Ubuntu 16.04 environment.

1) *E2E NS Deployment*

We validate our implementation by conducting an E2E network slicing experiment in which we: customize the VNF and NS descriptors (VNFD and NSD, respectively) to match our infrastructure configuration (i.e., network and compute services); install srsLTE; deploy the NS on the testbed; initialize the eNodeB and Evolved Packet Core (EPC); and successfully offer network service. After that, our E2E framework monitors the resource utilization, and performs dynamic resource allocation when deemed necessary, which is taken care of by the policy manager in OSM.

Our initial approach utilized Juju hooks to automate the installation and deployment of srsLTE in any compatible OS image. However, this approach resulted in a lengthy LTE installation time, measured to be ~890s. Knowing this, we decided to sacrifice some degree of flexibility in order to minimize the deployment time, by creating a pre-loaded image with srsLTE and all its dependencies installed. Next, we wrote the VNF and NS descriptors, where we used the provided configuration files from OSM as a baseline. We then proceeded to instantiate a NS based on the srsLTE image we created and using the NSD we wrote.

Under the hood, the LCM and VCA perform the proper service instantiation steps and inform the RO, which communicates with the VIM, which invokes the network and compute services. The VIM first creates the network using the specifications from the NSD, then the VNFs representing our srsLTE solution are created based on the instructions given in the VNFD. After the new NS instance is up and reachable, it starts bringing up the LTE system. First, the EPC with all the constituent components is brought up. Then the USRP is contacted and the eNodeB waits for a connection with the home subscriber server (HSS). After several seconds, the LTE system is fully functional and ready to accept users. In order to connect a UE, we implemented srsUE in a separate compute node from the srsLTE, and for the RF we used a second USRP. After starting the UE, it searches for a cell and starts the attachment process. Then, it obtains an IP address from the network that was established by the EPC. We test the implementation by pinging over the LTE link.

We have automated the described procedure and carried out 25 repetitions of the experiment. Fig. 4 shows the distribution of the amount of time it took to enable E2E network services. The few outliers (around 54 and 75s) are due to changes in the underlying server for the NS. When a server first hosts a VNF, it fetches the image from our centralized image service. Despite that, the average time is still ~47s, with an average of 19.5s belonging to the time need to start the EPC, eNodeB and attach a UE. The evaluation shows that the NS deployment time for the option with pre-loaded srsLTE image is almost 20 times faster than carrying out the installation of srsLTE, measured to be ~910s. This is a considerable difference that justifies our trade-off between flexibility and agility.

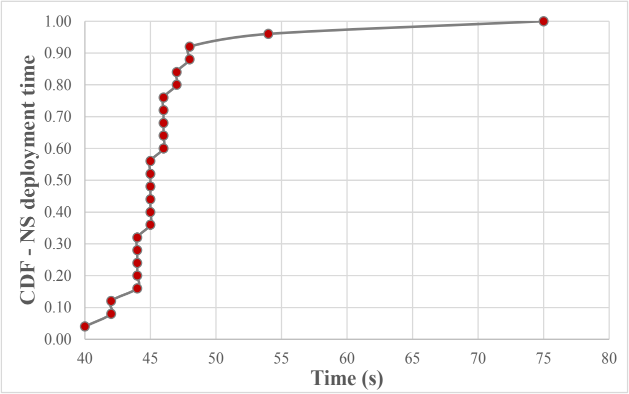


Fig. 4. Agile E2E NS deployment.

2) *SDN-Enabled Traffic Shaping*

SDN enables the ability to custom-tailor the network traffic types with respect to their QoS parameters. For our proof of concept, we designed a system which extends the lightweight srsLTE EPC to include traffic shaping, a method of rate limiting without dropping the excess packets. At the behest of our SDN controller, a programmable software switch [22] was used to enforce token bucket QoS policies in the created network service by changing the maximum bit rate (MBR) of the LTE system. The switch also enables redirecting different flows into queues with different policies.

Fig. 5 shows a box plot for the performance of TCP and UDP traffic for two service classes: one capped at 10Mbps and the other at 20Mbps. This can be easily extended to include more granular QoS with lower bit rates. It can be seen that UDP showed constant throughput in both cases, with an average close to the defined values, 9.93Mbps and 19.9Mbps respectively, and the median exactly at the service bit rates. Due to the wireless medium with high packet loss and frequent retransmissions, it can be noticed that the average throughput for TCP deteriorated for the 20Mbps class, where the average was 16.61Mbps, and the median was 18.45Mbps.

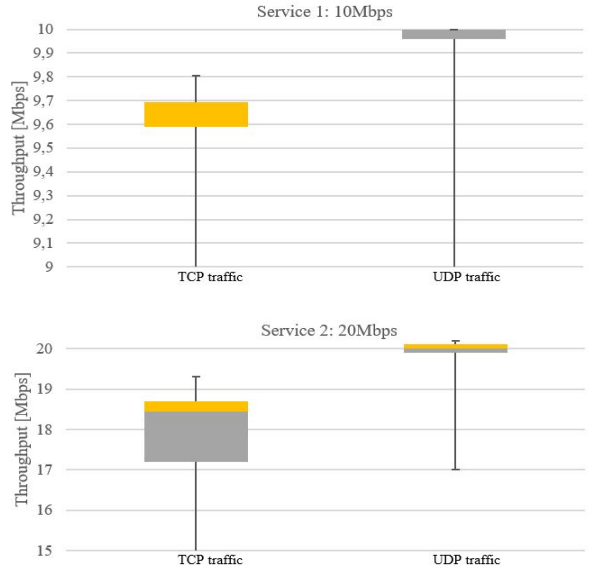


Fig. 5. SDN-enabled traffic shaping for 5G services.
The long TCP lower tail has been cut to save space.

As expected, the packet loss was lower for the 10Mbps class in which the average throughput for the TCP-based traffic was 9.11Mbps and the median was 9.49 Mbps. Thus, a major benefit from the traffic shaping is visible in the case of TCP-based flows where adapting to lower throughput rates was used in order to reduce retransmissions without exacerbating traffic congestion. The results prove that the ability to provide dynamic traffic shaping enables fine-grained control over the services and users, and improves the overall performance.

IV. Conclusion and future work

E2E network slicing is a paradigm brought by network virtualization and softwarization technologies to enable 5G. This paper addresses the design and implementation of an E2E network slicing framework on a multi-region and multi-tenant testbed.

Our evaluation results show that NS deployment through our logically centralized framework, including the creation and instantiation of all the constituent components, takes less than a minute when the proper design choices are made. We have also shown that the SDN-enabled system can be utilized to customize network services with specific QoS parameters, or adjust it based on the network conditions.

Future work will explore machine learning (ML) models for traffic prediction to enable informed and agile resource allocation, as well as support for other VIM types and virtualization technology (i.e. containers). It will also expand upon the area of SDN in order to enable completely programmable and adaptable network, with fine-grain QoS and access control.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong and J. C. Zhang, "What Will 5G Be?," *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 6, pp. 1065-1082, 2014.

[2] ETSI, "Network Functions Virtualisation (NFV)," 2017. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper_5G.pdf. [Accessed 1 April 2019].

[3] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proceedings of the IEEE, vol 103, no. 1, pp. 14-76*, 2015.

[4] Wireless World Research Forum, "End to End Network Slicing," 2017. [Online]. Available: https://www.wwrf.ch/files/wwrf/content/files/publications/outlook/White%20Paper%203-End%20to%20End%20Network%20Slicing.pdf. [Accessed 10 Oct. 2018].

[5] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration," 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf. [Accessed 1 April 2019].

[6] Open Source MANO, [Online]. Available: https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE_Documentation. [Accessed 16 February 2019].

[7] Juju, 28 March 2019. [Online]. Available: https://jujucharms.com/.

[8] L. Gavrilovska, V. Rakovic, A. Ichkov, D. Todorovski and S. Marinova, "Flexible C-RAN: Radio Technology for 5G," in *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Nis, 2017.

[9] Prometheus. [Online]. Available: https://prometheus.io/. [Accessed 27 March 2019].

[10] Grafana. [Online]. Available: https://grafana.com/grafana. [Accessed 27 March 2019].

[11] J.-M. Kang, T. Lin, H. Bannazadeh and A. Leon-Garcia, "Software-Defined Infrastructure and the SAVI Testbed," in *Testbeds and Research Infrastructure: Development of Networks and Communities. TridentCom 2014. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 137. Springer, Cham*, 2014.

[12] T. Lin, B. Park, H. Bannazadeh and A. Leon-Garcia, "Deploying a Multi-Tier Heterogeneous Cloud: Experiences and Lessons from the SAVI Testbed," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, 2018.

[13] B. Park, T. Lin, H. Bannazadeh and A. Leon-Garcia, "JANUS: Design of a software-defined infrastructure manager and its network control architecture," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, 2016.

[14] "OpenStack Queens," 28 March 2019. [Online]. Available: https://www.openstack.org/software/queens/.

[15] B. Nogales, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso and A. Azcorra, "Experimentation, Design and Deployment of an Open Management and Orchestration Platform for Multi-Site NFV," *IEEE Communications Magazine,* vol. 57, no. 1, pp. 20 - 27, 2019.

[16] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger and L. Dittmann, "Cloud RAN for Mobile Networks—A Technology Overview," *IEEE Communications Surveys & Tutorials* , vol. 17, no. 1, pp. 405-426, 2014.

[17] [Online]. Available: https://www.ettus.com/all-products/x310-kit/. [Accessed 29 March 2019].

[18] V. Rakovic, A. Ichkov, S. Marinova, D. Todorovski, V. Atanasovski and L. Gavrilovska, "Dynamic Virtual Resource Allocation in Virtualized multi-RAT Cellular Networks," *Springer journal,* pp. 1-16, 2017.

[19] [Online]. Available: https://github.com/srsLTE/srsLTE. [Accessed 29 March 2019].

[20] Z. Geng, X. Wei, H. Liu, R. Xu and K. Zheng, "Performance analysis and comparison of GPP-based SDR systems," in *7th IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE)*, Xi'an, 2017.

[21] F. Gringoli, P. Patras, C. Donato, P. Serrano and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," *IEEE Wireless Communications,* vol. 25, no. 5, pp. 10-15, 2018.

[22] Open vSwitch, [Online]. Available: http://docs.openvswitch.org/en/latest/intro/what-is-ovs/. [Accessed 5 April 2019].