Scaling Network Slices with a 5G Testbed: A Resource Consumption Study

Tolga O. Atalay*, Dragoslav Stojadinovic†, Angelos Stavrou*†, Haining Wang*
*Department of Electrical and Computer Engineering, Virginia Tech, USA
†Kryptowire, LLC, McLean, VA, USA
Email: tolgaoa@vt.edu, dstojadinovic@kryptowire.com, angelos@vt.edu, hnw@vt.edu

***Abstract*—The next generation of networks will be utilized by multiple industry verticals with different service requirements on top of a common infrastructure. Through network function virtualization (NFV), the 5G core and Radio Access Network (RAN) functions are now implemented as virtual network functions (VNFs) on commercial off-the-shelf (COTS) hardware. The use of virtualized micro-services to implement these 5G VNFs enables end-to-end logically isolated network slices on a large scale. In this paper, we seek to measure, analyze, and understand the limits of 5G micro-service virtualization when using lightweight containers to realize different network slicing models with different service guarantees. Our deployment consists of the OpenAirInterface (OAI) core and a simulated RAN in a containerized setting to create a universally deployable testbed. We perform stress tests on individual VNFs and create network slicing models applicable to real-life scenarios. Our analysis captures the increase in compute resource consumption of individual 5G VNFs during various core network procedures. Furthermore, using different network slicing models, we are able to see the progressive increase in resource consumption as the service guarantees of the slices become more demanding. The framework created using this testbed is the first to provide such analytics on lightweight virtualized 5G core VNFs with large scale end-to-end connections.**

***Index Terms*—5G testbed, 5G core, network slicing, network functions virtualization (NFV), OpenAirInterface (OAI)**

I. Introduction

In the last few years, mobile networks have quickly evolved into an ecosystem that will soon be able to accommodate applications and use cases with a very diverse set of Quality of Service (QoS) guarantees over the same physical infrastructure. To keep up with the growing versatility of use cases in the next generation of wireless networks, the concept of network slices was created. A network slice can be described as a virtualized logical network with customer-specific QoS utilizing a set of shared underlying physical networking and computing resources. Different slices can support a variety of vertical use cases alongside enhanced mobile broadband (eMBB), ultra reliable low latency communication (URLLC), massive internet of things (mIoT) and vehicle-to-everything (V2X) communication.

This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement number HR001120C0155. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

In legacy mobile networks, core services were implemented in proprietary hardware which made them difficult to manipulate after the initial deployment. Utilizing network functions virtualization (NFV) and software-defined networking (SDN) as building blocks, 5GC networks strive to implement all the core services as VNFs on commercial off-the-shelf (COTS) hardware. A network slice is formed by service chaining a series of virtual network functions (VNFs) which have been selected to accommodate the performance requirements of the users allocated to them.

Three network slices are depicted in Fig. 1, where users with different QoS requirements are allocated to network slices with optimally selected VNFs that have been instantiated inside containers around different locations inside the network. For users requiring low latency, data plane anchors such as the Session Management Function (SMF) and the User Plane Function (UPF) are deployed in the edge network while the more management oriented functions are instantiated in the distributed or central cloud. This virtualized environment

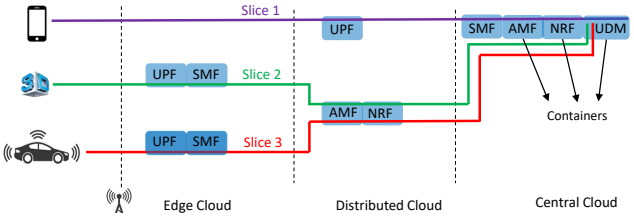


Fig. 1: Network Slice Specific VNF Deployment

provides a high degree of flexibility to VNF instantiation leading to finer control over compute resources. It is important to understand the compute footprint of these VNFs in containerized environments to gain a better intuition into how they will scale in a real-life deployment.

5G testbeds have been largely focused on the radio access network (RAN) and have neglected to analyze the impact of a virtualized core network [1]. Recently, open-source solutions such as OpenAirInterface (OAI) [2] have developed enough for a feature-rich implementation. Our goal is to make use of these existing 5G solutions to assess the compute resource consumption of a lightweight virtualized 5GC network. Our contributions are given below.

- Firstly, we use the OAI 5GC in a *large-scale* con-

tainerized deployment, namely the Access and Mobility Management Function (AMF), the SMF, the Network Functions Repository Function (NRF), Home Subscriber Server (HSS) and the Serving Gateway (S-GW) with 5G features, which is the legacy variant of the UPF for 5G.
- Secondly, using the gNBSIM [3] RAN simulator, we deploy mass end-to-end connections between the RAN and the 5GC to observe the compute resource consumption behaviour of the VNFs in various stress tests.
- Finally, we create different slicing models applicable to real-life scenarios to provide insight into scalability implications surrounding network slicing by observing the trade-off between performance, deployment time and resource consumption.

II. Related Work

This section summarizes the existing open-source projects and frameworks dedicated towards building a 5G network as well as the various testbed studies that have spawned revolving around them.

A. 5G RAN and Core Open-source Projects

Numerous projects have dedicated themselves to the implementation of 5G standardization. The leaders in this effort are summarized in Table I.

TABLE I: Open-source 5G Developments

	RAN	Core
srsRAN [4]	formerly srsLTE, has a stable LTE RAN with no 5G components	only EPC with no indications towards a future 5GC
Open5GS [5]	Used with RAN simulators	Service-based interfaces up to Rel.16
free5GC [6]	Used with RAN simulators	most service-based interfaces between VNFs implemented
OAI [2]	Work in progress 5G gNB and UE	Fundamental components of Rel.16

Being the only other project with a RAN component, srsRAN currently only supports the LTE eNB and UE with an EPC. They have made no indications towards the development of a 5GC. Projects like Open5GS and free5GC have explicitly focused on the core network development without an in-house RAN component.

OAI is currently the only project with a 5G gNB and UE implementation that also has a 5GC network with all the fundamental services. Their 5G RAN solutions, while available to use and run, are still in early stages and are being actively developed [7].

B. Open Frameworks and Projects

In addition to the projects that develop the RAN and core network services, there are efforts within the 5G community to enhance the interactions with these components. Most well-known projects are summarized in Table II.

TABLE II: Open-source 5G Frameworks

O-RAN [8]	Linux Foundation	Disaggregated RAN with software-define control over radio resource control
Open Network Automation Platform (ONAP) [9]	Linux Foundation	5G customized orchestration platform with built-in network slicing management
Software Defined (SD)-RAN [10]	Open Network Foundation	OAI augmentation project for O-RAN compatible RAN components
MOSAIC5G [11]	OAI Alliance	OAI sub-project including platforms such as FlexRAN and Kube5G and O-RAN integration

O-RAN aims to create a disaggregated RAN by means of employing a functionality split among the central, distributed and radio units. O-RAN uses an SDN controller, referred to as the RAN intelligent controller (RIC) for VNFs in the RAN. This allows for a softwarized control of radio resources, ultimately allowing for an optimal resource scheduling mechanism as well as other policy adjustments and load balancing.

Complementing O-RAN from a management perspective, there is ONAP, which is a highly modular MANO tool, specifically developed for 5G NFV with an intrinsic 3GPP management system for network slicing. Serving as an NFV orchestrator and VNF manager, ONAP can be integrated with multiple virtual infrastructure managers.

Given the growing popularity of O-RAN in the industry, different projects have spawned that aim to develop compatible RAN components with it. SD-RAN and MOSAIC5G are such projects which aim to augment OAI with an agent that can communicate with the O-RAN RIC.

C. Utilization of Testbeds

A testbed is essential for gathering high fidelity measurements in 5G mobile networks. This section presents some of the more recent 5G testbed prototypes.

Studies dedicated to analyzing the core and network slicing either use legacy implementations, disregard virtualization or simply leave the implementation at a proof-of-concept. In [12] such a proof-of-concept has been developed to carry out experiments with network slicing in virtualized environments, however the findings are at a prototyping stage and no measurement or analytics is provided. The authors of [13] have carried out the deployment of the OAI EPC and the LTE RAN as opposed to the recent 5G components in a virtualized environment, however the work only presents a proof-of-concept and no findings towards the resource consumption of this kind of a setup. A similar legacy OAI testbed is used in [14], where the authors have deployed the OAI EPC with an external SDN controller to test out the performance of a dynamic network slicing scheme.

A study that is closer in spirit to our goal, provides measurements on resource consumption in the core network when creating individual network slices made up of the free5GC VNFs with a simulated RAN connection [15]. The VNFs are instantiated inside individual OpenStack VMs instead of lightweight Docker containers which creates a resource-heavy virtualization environment as opposed to a more lightweight approach with containers.

The work in [16] uses different virtualization techniques such as VMs and Docker containers as well as bare metal to deploy the EPC of Open5GCore with an LTE RAN. Their goal is to analyze the effect of a virtualized core network on machine-type communication (MTC) traffic rather than the resource consumption of the core network functions themselves.

Very recently conducted set of experiments in [1] showcase the current capabilities of the OAI RAN. The authors have deployed the OAI RAN on SDRs with Open5GCore as the core connection. However, they only provide measurements and analysis regarding the RAN with no consideration of the resource consumption of the core VNFs.

These are some of the more relevant examples of 5G testbed implementations. For a more comprehensive and systematic review of the most recent 5G testbeds, reader is directed to [17]. To the best of our knowledge, our work is the first to create a containerized 5GC network at a large scale deployment to assess the resource consumption of a lightweight virtualized core and analyze different slicing models.

III. Containerized Deployment

For large scale experimentation we chose to use a simulated RAN component that is compatible with the 5G core VNFs. This allows us to create scalable high-stress environments which would have been hard to achieve using SDRs. Hence, we build a containerized deployment of the gNBSIM RAN simulator and the OAI 5GC network, where the core network functions along with a gNB and UE are deployed to form an end-to-end connection between a user and the network. This step forward with containerization creates a core network that is easy to deploy and manage. Furthermore, the low overhead of the containerized environments allows us to replicate these deployments for large scale experimentation which would have been very difficult to do in baremetal and inconvenient using VMs.

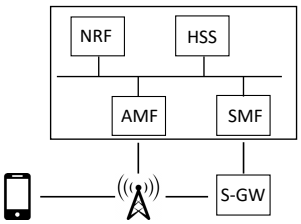


Fig. 2: OAI Architecture

The end-to-end logical architecture of the setup is given in Fig. 2. The developed core network functions are described below.

- MySQL, which is the HSS in LTE and Unified Data Repository (UDR) in 5G.
- NRF, the metadata database and communication hub the VNFs register to when joining the network.
- AMF, the primary point of contact with the UE in the core and the main orchestrator for processing and forwarding non-access stratum information to other relevant VNFs.
- SMF, the anchor point for the packet data unit (PDU) session in the control plane.
- S-GW, the legacy network function from LTE augmented with 5G features of the UPF as the user plane anchor point for the PDU session.

All core VNFs as well as gNBSIM are deployed inside a separate Docker container as shown in Fig. 3.

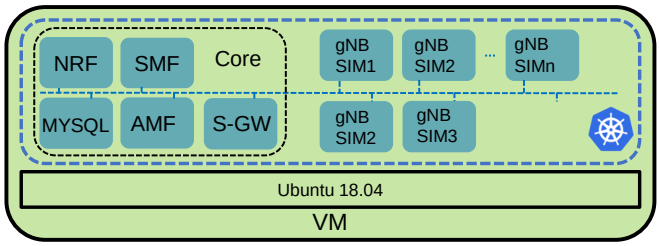


Fig. 3: Containerization

The size of the Docker images for the core and RAN is given in Table III.

TABLE III: Size of Docker Images (MB)

MYSQL	NRF	AMF	SMF	S-GW	gNBSIM
695	242	333	247	228	147

IV. Experimental Setup

Our testbed is composed of two Dell Precision 7920 Tower servers with:

- 2 x Intel Xeon Gold 5218R 2.1GHz CPUs,
- 512GB RAM,
- 1TB disk space,
- 2 x 1G network interface cards,

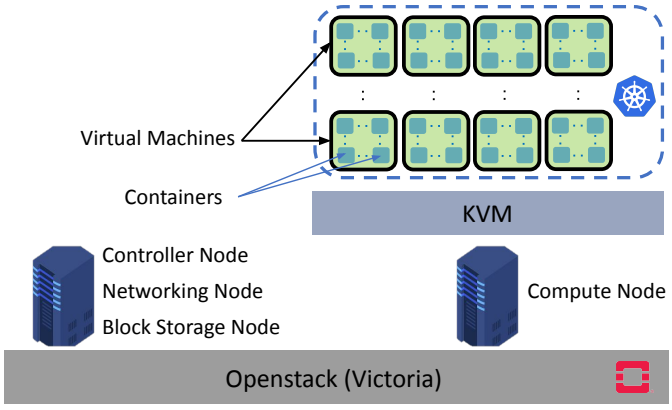


Fig. 4: Infrastructure Setup

The setup is depicted in Fig. 4. We have set up OpenStack Victoria on a two-server infrastructure with one server as the controller, network and block storage node and the other as the compute node. The compute node uses KVM hypervisor for virtualization and has a total of 80 vCPUs. On top of the compute node, 16 VMs are instantiated for the purpose of creating a High-Availaibility (HA) Kubernetes cluster. Each VM has the Ubuntu 18.04 OpenStack server image. The VMs used for the cluster are given in Table IV.

TABLE IV: HA Kubernetes Cluster VMs Configuration

Node	Instances	vCPUs	RAM (GB)	Disk (GB)
Control	3	2	8	30
NFS	1	2	8	80
Worker	12	6	16	50

The cluster uses 3 control plane and 12 worker VMs and a network file system (NFS) node for the persistent volumes.

V. Performance Evaluation

In this section we perform stress tests on the VNFs in the OAI 5GC by forcing high utilization of their respective processes. This provides insight into the resource consumption of the 5GC network and what to expect from a real-life deployment when allocating resources to specific VNFs. Next, we create different VNF sharing schemes for network slicing models and show how it affects the trade-off between performance and resource-consumption.

A. VNF Stress Tests

For the NRF, we continuously register VNFs and observe the effect of the registration process as well as the effect of maintaining a growing pool of VNF profiles. For the AMF, users are registered consecutively, where AMF needs to record their authentication profiles. Similarly for the SMF, PDU sessions follow the same cycle and the session context data that is stored grows with each user. For the UPF, once the users are registered and their PDU sessions are ready, traffic is generated starting with the first user down to the last user with a delay between instantiations.

In Fig. 5a, during the VNF registration procedure, the NRF processes the metadata for each VNF and maintains it. Regular updates are performed to maintain the accuracy of the information so that it can be broadcast to the other VNFs that wish to consume relevant services. Maintaining the update procedure increases the strain on the NRF and we observed general instability and crashes after 340 registered VNFs. It is possible to see an overall increase in the CPU consumption. As more VNFs are registered, the NRF needs to perform more updates on the profiles it maintains. The memory consumption is stable throughout this lifecycle since no additional memory is consumed other than the initial base image.

In Fig. 5b, the gNBSIM RAN simulator is deployed with gNB and UE pairs and traffic is generated for these users through a single UPF from an externally configured data network node (DNN). All the user traffic is tunneled through a single UPF interface at the same time. For now we are simply using the iperf3 traffic pattern. While this scenario is not realistic as it will hinder QoS significantly in a practical deployment (shown in Fig. 6c for model 1), it is performed here to investigate the resource consumption capacity of the UPF. At the peak of CPU utilization there are 130 active sessions. As the PDU sessions are terminated, the CPU utilization of the UPF decreases while memory consumption is constant throughout the entire process.

In Fig. 5c, the AMF authenticates both with the gNB and the UE during each registration process. First, the gNB uses an operator key for authentication after which the user associated with that gNB authenticates with the network using their own key. The CPU utilization shows sudden increases responding to incoming registration requests. As it registers each user, AMF keeps a record of the operator and user profiles which increases its memory consumption as the number of users grows. To avoid duplicate users, each user and gNB pair has a unique configuration. Instability was observed at more than 130 gNBSIM pairs.

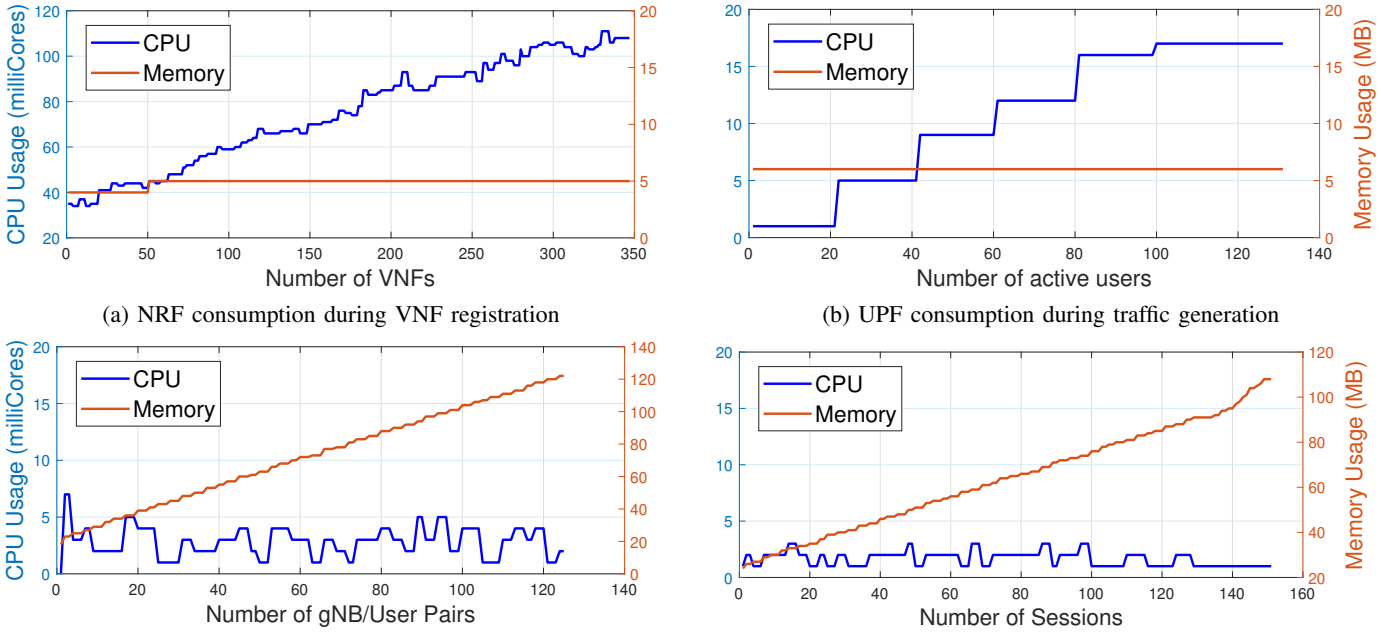
For the SMF in Fig. 5d, the setup of each PDU session is carried out one at a time which only affects CPU utilization for short intervals similar to the AMF. Initially an SMF is designated for each user after registration. Following this, a PDU session is created for the user with a chosen UPF where the SMF manages the control plane. After setting up the PDU session, SMF maintains the context data of the subscribed users which is why it consumes an increasing amount of memory with each user.

B. Network Slicing Configurations

For comparing potential real-life deployments with different network slicing configurations, we used various models where the VNFs could be shared among a fixed number of users. Depending on the model, some VNFs are shared by a higher number of users and others by fewer users. The more centralized VNFs like NRF and AMF are shared at a larger scale because they are not a part of the specific PDU sessions of a user. On the other hand, the session anchors like the SMF and UPF, are shared at a smaller scale given that any congestion in these VNFs will affect the QoS of the user. Table V shows how many users share a given VNF type and the total number of instances that are deployed for different slice configurations. The number of users in each case is fixed to 80.

There are a total of five models each of which pertains to a use case.

- **Model 1**: one-slice-fits all model as a benchmark.
- **Model 2**: NRF is globally shared in a given domain and the remaining VNFs are shared at a larger scale. This model can be associated with a traditional eMBB service type where the users have no specific requirements.
- **Model 3**: NRF is shared at a large scale while the AMF is shared at a smaller scale. SMF and UPF are shared by two users. This type of deployment is suitable for users requiring stringent QoS requirements.



(a) NRF consumption during VNF registration

(b) UPF consumption during traffic generation

(c) AMF consumption during user authentication

(d) SMF compute consumption during PDU session setup

Fig. 5: VNF Stress Tests

TABLE V: Network Slice Configurations

Model	NRF		AMF		SMF/UPF	
	Instances	Shared Users	Instances	Shared Users	Instances	Shared Users
1 - Benchmark Case	1	80	1	80	1	80
2 - General eMBB Service	1	80	10	8	10	8
3 - Specialized QoS Agreement	5	16	20	4	40	2
4 - High Level Management	10	8	10	8	20	4
5 - E2E Logical Isolation	40	2	40	2	40	2

- **Model 4**: NRF and AMF are centralized even further to facilitate a higher level of management by serving larger groups of slices while the SMF and UPF are shared at a smaller scale to preserve PDU session QoS. While model 3 is considered as a specialized user-specific service type, model 4 is the generalized slice-service type that can be used for a wider variety of users.
- **Model 5**: NRF and AMF are also implemented at a slice-specific level to maintain end-to-end logical slice isolation. This scenario applies to users with high-security requirements.

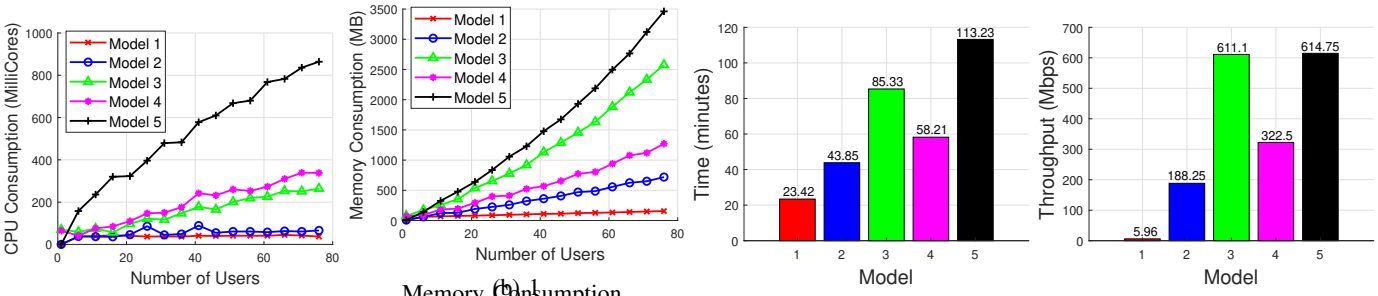
All the users for one slice are instantiated once all the VNFs for that slice are ready. When one of the VNFs in that slice fills its quota, new VNFs are created to accommodate the additional users before new users are deployed.

The CPU consumption of these slice configurations is given in Fig. 6a and memory consumption in Fig. 6b. Additionally, the deployment time is displayed in Fig. 6c to gain a sense of the real-life feasibility of each model. Finally, Fig. 6d shows the throughput of the PDU sessions for each model. Individual DNNs are used for each user to make sure that there is no bottleneck in the data network so that the congestion in the 5G VNFs can be accurately monitored.

For model 1, with the one-slice-fits-all approach the resource consumption is very low at the expense of high congestion in the VNFs. It has a short deployment time given that each VNF is deployed only once. On the other hand, due to the single PDU session anchors, the users experience very low throughput.

For model 2, the NRF is globally shared, which means that all the VNFs are recorded in a single location. The AMF, SMF and the UPF are shared at a large scale which has a visible impact on memory consumption compared to model 1. The creation of multiple PDU session anchors lowers the congestion to yield a slightly better throughput. However, it is possible to see that deployment time approximately doubles compared to model 1, implying that even recycling large slices is far more time-costly.

For model 3, with the user-specific QoS accommodation model in the PDU session, the SMF and UPF are shared at a much smaller scale. The high amount of virtualization overhead associated with the base image of each SMF and UPF causes much higher resource consumption. With the PDU session anchors only serving two slices, the performance increases drastically at the expense of once again doubling deployment time.



(a) CPU Consumption

(b) Memory Consumption

(c) Deployment Time

(d) Average Throughput

Fig. 6: Network Slice Models Analysis

In model 4, the increased number of NRFs leads to higher CPU utilization as evidenced by the CPU consumption of NRF in Fig. 5a. However, since there are fewer of the other VNFs, the memory consumption is lower than that of model 3. With each PDU anchor serving four slices, it is possible to observe performance degradation compared to model 3.

Finally, model 5 promotes end-to-end isolation. While this will create more secure slices, it comes at the cost of deploying a higher number of instances which significantly increases both CPU and memory consumption. The PDU session configuration is identical to model 3, however, the added cost of deploying the management-related VNFs as slice-specific has a drastic impact on deployment time.

Overall the CPU utilization in Fig. 6a of the slice models demonstrates periodic surges coinciding with the triggering of VNF functionalities. The series of VNF registrations to the NRF(s) contribute to a steady increase as evidenced by the individual statistics in Fig. 5a. Additionally, it is possible to observe steeper increases in CPU consumption with stricter sharing models as UE/gNB pairs are created. This results in instantaneous utilization by the AMFs and SMFs shown in Fig. 5c and Fig. 5d to occur at a larger scale. The utilization amount is higher with more demanding models given that there are more VNFs.

VI. Conclusion

The concept of network slicing has become a necessity with the increasingly various set of requirements. In this paper, we provide the reader with an intuition regarding the real-life scalability implications of network slices and individual 5GC VNFs. we demonstrate our containerized deployment of the OAI 5GC network and the gNBSIM RAN simulator in a Kubernetes environment. We utilize an environment with abundant compute resources to replicate this deployment at a large scale on top of Openstack. To provide compute resource consumption insight, various stress tests were performed on the 5G VNFs using a large number users. Such tests allowed us to see how these VNFs can react in a real-life scenario. Finally, we create different network slice configurations where VNFs were shared among slices to demonstrate the resource consumption of different service provisioning settings. These include the generic eMBB scenario, a user-specific QoS profiling, a less stringent slice-specific setup and end-to-end slice isolation for services requiring increased security.

In the future, we will further enhance this testbed by modelling more complex traffic patterns incoming from even a larger set of users. Different traffic patterns will have different results on the UPF resource consumption. Furthermore, we plan to integrate O-RAN compatible emulated RAN nodes to look into security related issues that accompany this virtualized deployment.

References

[1] M. Vilakazi, C. R. Burger, L. Mboweni, L. Mamushiane, and A. A. Lysko, "Evaluating an Evolving OAI Testbed : Overview of Options, Building tips, and Current Performance," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2021, pp. 818–827.

[2] "OpenAirInterface 5G software alliance for democratising wireless innovation," https://openairinterface.org/, (Accessed on 09/08/2021).

[3] "gNBSIM Gitlab," https://gitlab.eurecom.fr/kharade/gnbsim, (Accessed on 09/24/2021).

[4] "srsRAN Your own mobile network," https://www.srslte.com/, (Accessed on 09/08/2021).

[5] "Open5GS Open source project of 5GC and EPC (release-16)," https://open5gs.org/, (Accessed on 09/08/2021).

[6] "free5GC," https://www.free5gc.org/, (Accessed on 09/08/2021).

[7] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-art and the Road Ahead," *Computer Networks*, vol. 182, p. 107516, 2020.

[8] "O-RAN ALLIANCE," https://www.o-ran.org/, (Accessed on 09/08/2021).

[9] "ONAP," https://www.onap.org/, (Accessed on 09/08/2021).

[10] "ONF SD-RAN," https://opennetworking.org/sd-ran/, (Accessed on 09/10/2021).

[11] "Mosaic5G," https://mosaic5g.io/, (Accessed on 09/10/2021).

[12] A. Esmaeily, K. Kralevska, and D. Gligoroski, "A Cloud-based SDN/NFV Testbed for End-to-End Network Slicing in 4G/5G," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 29–35.

[13] B. Dzogovic, V. T. Do, B. Feng, and T. van Do, "Building Virtualized 5G Networks Using Open Source Software," in *2018 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, 2018, pp. 360–366.

[14] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "Dynamic Network Slicing for 5G IoT and eMBB Services: A New Design with Prototype and Implementation Results," in *2018 3rd Cloudification of the Internet of Things (CIoT)*, 2018, pp. 1–7.

[15] C.-W. Liao, F. J. Lin, and Y. Sato, "Evaluating NFV-enabled Network Slicing for 5G Core," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 401–404.

[16] H.-C. Chang, B.-J. Qiu, C.-H. Chiu, J.-C. Chen, F. J. Lin, D. de la Bastida, and B.-S. P. Lin, "Performance Evaluation of Open5GCore over KVM and Docker by using Open5GMTC," in *NOMS 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–6.

[17] A. Esmaeily and K. Kralevska, "Small-Scale 5G Testbeds for Network Slicing Deployment: A Systematic Review," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.