

Evaluating Dedicated Slices of Different Configurations in 5G Core

Yu-Herng Chai, Fuchun Joseph Lin*

Department of Computer Science, College of Computer Science, National Yang Ming Chiao Tung University, Taiwan

Email: chris60303@gmail.com, *fjlin@nycu.edu.tw

How to cite this paper: Chai, Y.-H. and Lin, F.J. (2021) Evaluating Dedicated Slices of Different Configurations in 5G Core. *Journal of Computer and Communications*, 9, 55-72.

<https://doi.org/10.4236/jcc.2021.97006>

Received: June 28, 2021

Accepted: July 27, 2021

Published: July 30, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Network slicing is one of the most important concepts in 5G networks. It is enabled by the Network Function Virtualization (NFV) technology to allow a set of Virtual Network Functions (VNFs) to be interconnected to form a Network Service (NS). When network slices are created in 5G, some are shared among different 5G services while the others are dedicated to specific 5G services. The latter are called dedicated slices. Dedicated slices can be constructed with different configurations. In this research, dedicated slices of different configurations in 5G Core were evaluated in order to discover which one would perform better than the others. The performance of three systems would be compared: 1) Free5GC Stage 2 with each dedicated slice consisting of only UPF; 2) Free5GC Stage 3 with each dedicated slice consisting of only UPF; 3) Free5GC Stage 3 with each dedicated slice consisting of both SMF and UPF in terms of their registration time, response time, throughput, resource cost, and CPU utilization. It is shown that not one of the above systems will always be the best choice; based on the requirements, a specific system may be the best under a specific situation.

Keywords

Network Function Virtualization (NFV), Next Generation Networking, Resource Management, Network Slice, Dedicated Slice, Free5GC

1. Introduction

5G networks aim to support lots of applications that are characterized by diverse performance requirements. Network slicing [1] is the key technology enabler to achieve this target. Much research has been done on how to create network services using network slicing [2] [3]. However, very few researches show how different configurations of dedicated slices will affect the performance of 5G core.

In this research, two different configurations of dedicated slices in free5GC Stage 3 will be explored [4]. On one configuration, we will create a common slice that consists of NRF (Network Repository Function), AMF (Access and Mobility Management Function), UDR (Unified Data Repository), PCF (Policy Control Function), UDM (Unified Data Management), NSSF (Network Slice Selection Function) and AUSF (Authentication Server Function) and three dedicated slices of different configurations that consist of SMF (Session Management Function) and UPF (User Plane Function). On the other configuration, the common slice will consist of all control plane VNFs including NRF, AMF, UDR, PCF, UDM, NSSF, SMF and AUSF while each dedicated slice will consist of only UPF.

Furthermore, the difference between free5GC Stage 2 and Stage 3 will be compared. Note that in free5GC Stage 2, NSSF is not utilized for the selection of a specifically dedicated slice to serve the requesting UE. On the other hand, free5GC Stage 3 utilizes NSSF according to the 3GPP standards; its NSSF would give the list of slice candidates to AMF, so AMF can choose the best one of them to provide the service to the requesting UE.

Thus overall, the performances of three systems will be compared: 1) Free5GC Stage 2 with each dedicated slice consisting of only UPF; 2) Free5GC Stage 3 with each dedicated slice consisting of only UPF; 3) Free5GC Stage 3 with each dedicated slice consisting of both SMF and UPF. Note that all the slices under our experiment will be provisioned before the system starts. There will be no support for the dynamic creation of dedicated slices.

A traffic generator will be used to send packets toward a specific 5G slice. Our proposed design will make sure all traffic will follow the flows defined in 3GPP standards. The performances of the three aforementioned systems in terms of their registration time, response time, throughput, memory, and CPU utilization will be compared. We expect that none of the above systems will always be the best choice. Based on the requirements, a specific system may be the best under a specific situation.

To our best knowledge, this research is the first to evaluate different configurations of dedicated slices in 5G and investigate their impact on the overall system performance. This is the major contribution of this research.

The rest of this paper is organized as follows. Section II presents the background and related work, focusing on ETSI (European Telecommunications Standards Institute) NFV MANO (MANagement and Orchestration) framework and the open sources used in this research. Section III describes the different configurations of dedicated slices in 5G Core. Section IV shows the implementation and evaluation of these dedicated slices. Finally, Section V concludes this paper and shows potential future work.

2. Background and Related Work

In order to compare different configurations of dedicated slices, we build our testbed based on the ETSI NFV [5] MANO framework [6] [7], NYCU free5GC,

OpenStack [8] and Tacker [9]. In this section, we will explain each of the above technologies and present other related researches on 5G dedicated slices, 5G core slicing and NSSF.

2.1. ETSI NFV MANO Framework

ETSI NFV MANO is an architectural framework that utilizes virtual resources to create Network Services (NSs). Each NS can contain one or more Virtual Network Functions (VNFs).

VNF can be deployed and executed on the virtual environment such as virtual machine (VM) or container. VNF utilizes the underlying virtual resources from the NFVI for its execution. Multiple VNFs can be chained together to form a network service.

As depicted in **Figure 1**, NFV Orchestrator (NFVO), VNF Management (VNFM) and Virtualized Infrastructure Managers (VIM) are three major components in the MANO architecture framework.

- **NFVO:** NFVO receives the NS instantiation-related resource descriptions from the OSS/BSS to onboard a network service (NS). An NS consists of a set of VNFs chained together via Virtual Links (VLs). NFVO manages the deployment of VNFs and VLs for an NS and orchestrates the underlying infrastructures to instantiate an NS. NFVO deploys VNFs through the VNFDs (VNF Descriptors) that are used to describe not only the behaviors of the VNFs but also the relationship between VNFs through VLs. Moreover, VNFD describes the instantiation parameters such as node types and node templates. The node templates specify important attributes such as VDU (Virtual Data Unit), internal VL and FIP (Floating IP). VDUs describe all the information about the VNF Components (VNFCs) including their names, images and flavors. VL provides the connections among VNFs. Floating IP allows us to control the instances from a remote site.

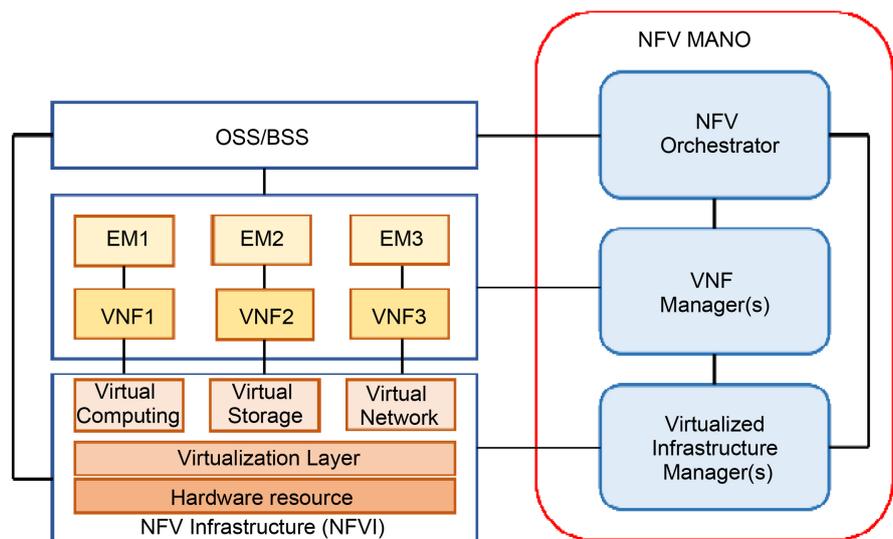


Figure 1. NFV MANO architecture.

- VNFM: It oversees the lifecycle management of VNF instances including creation, modification and termination. It plays the coordination and adaptation role between NFVI and Element Management System (EMS).
- VIM: It controls and manages the NFVI resources such as compute, storage and network resources. These resources are transformed from hardware resources to virtual resources through virtualization layer.

According to 3GPP, network slice consists of multiple network slice subnets such as core network, access network and transport network. NFV MANO plays an important role to map each network slice subnet to a network service.

2.2. Related Open Sources

First, we used NYCU free5GC which is an open-source project for 5G mobile core network. Some open sources claim to be 5G Core but still using the non-standalone configuration where EPC is used as the core network. On the other hand, NYCU free5GC is designed as a standalone 5G core network.

As shown in **Figure 2**, NYCU free5GC provides all 5G network functions and divides them into the control plane and user plane. In the control plane, there are NSSF, NRF, UDM, PCF, Network Exposure Function (NEF), AUSF, AMF, SMF, UDR and Application Function (AF). In the user plane, there is only one network function called UPF. The separation of user plane and control plane allows independent deployment and evolution of each. NYCU free5GC also modularizes the function design, which makes the deployment of its network functions under the NFV environment more flexible and efficient.

Its network functions are briefly described below:

- NSSF: Provides the list of available network slices and helps AMF to select the slice.
- NRF: Provides registration of network functions and records IPs of all network functions.
- UDM: Responsible for unified data management and manages the UDR.

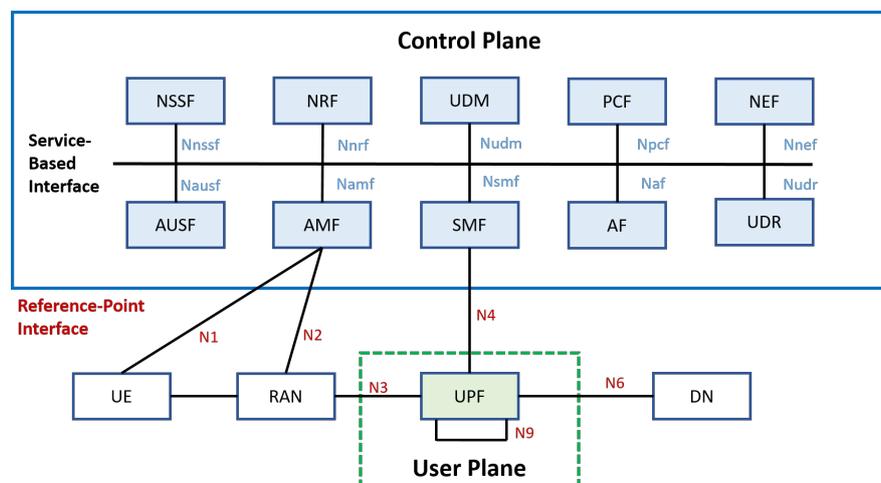


Figure 2. NYCU free5GC architecture.

- UDR: A database for storing all the information about UE.
- PCF: Provides policies and rules of control plane.
- NEF: Facilitates secure, robust, developer-friendly access to exposed network services and capabilities.
- AUSF: Provides authentication and authorization.
- AMF: Manages the new user's connection request. When there is a new UE trying to register to the core network, this UE will send a request to AMF.
- SMF: Manages the session between UPF and AN.
- AF: 5G Application Function.
- UPF: Responsible for packets forwarding.

In addition to free5GC, we also use MANO open sources including OpenStack [10] [11] and Tacker. OpenStack is an open-source cloud operating system which can virtualize resources including storage, compute and network. These virtualized resources are mostly deployed as Infrastructure-as-a-service (IaaS) in the cloud. OpenStack provides a dashboard which is a web-based interface. We can provision, manage and monitor our virtualized resources efficiently through this dashboard. Tacker is an OpenStack project which can provide the functionalities of NFVO and VNFM to configure, deploy, manage and orchestrate NSs and VNFs on an NFV infrastructure platform like OpenStack. Tacker APIs can be used by not only NFV orchestrator but also OSS/BSS to deploy VNFs.

2.3. Related Work

Dedicated slices are also called event slices because they are launched due to the occurrence of a special event such as a concert. Consequently, they are often with relatively short life cycles [12]. Below several related works on the design of 5G core slicing [13] are surveyed.

- In [14], potential challenges in 5G core slicing such as slice creation, slice management and security in network slicing were elaborated. Our system tackles these challenges by leveraging APIs provided by OpenStack and Tacker for slices creation and management. Also, to protect the VNFs in slices our system relies on the security group provided by OpenStack and Tacker.
- In [15], the modularization of 5G Core is identified as an important feature since this would allow independent evolution of its modules in the future.

Our system thus adopts NYCU free5GC which follows modularized functional design.

- Another important issue related to network slicing is how to guarantee hard isolation between slices [16]. Our system resolves this issue by utilizing the VM-based system architecture where the policy of strict no-resource-sharing between VMs is enforced.
- The KPIs (Key Performance Indicators) for network slicing based on ETSI NFV MANO architectural framework were defined in [17]. On the other

hand, the main concepts and principles of network slicing are such as the NFV, SDN and cloud technologies are well elaborated in [18]. The above two papers provide a comprehensive overview of network slicing.

The design of NSSF is also an important issue in our research where we need to select a slice from multiple available ones.

- The concept of IMSI based slice selection is proposed in [19] where a slice table was created first and NSSF would select the data plane slice based on User Equipment's (UE) IMSI. Our system also proposes to let NSSF select different kinds of slices based on the requests of different UEs.
- Network slicing selection matching model is designed in [20] where network slice registry and user request are used to assist the UE to select the network slice. Network slice registry is similar to the flow table in the OpenFlow protocol. Our proposed design follows similar ideas where the UE requests a network slice using Network Slice Selection Assistance Information (NSSAI).
- A concept called slice negotiation is proposed in [21] where an application/UE would negotiate with the serving network through Service Description Document (SDD). On the other hand, for our system before a slice is deployed in the VM-based environment, the VNFs need to be described in VNFDs (VNF Descriptor).

3. Design of Different Configurations in Dedicated Slices of 5G Core

In our 5G systems, we use OpenStack as our VIM and Tacker as our NFVO and VNFM to construct the MANO system. On the other hand, NYCU free5GC provides us a set of comprehensive 5G core network functions in order to support the network slicing of 5G core. Tacker is used to onboard and create free5GC VNFs on OpenStack through VNFD. Some VNFs are deployed in the common slice while the others in the dedicated slices.

Our goal is to construct an NFV MANO testbed for network slicing [22] of 5GC and evaluate the performances between different configurations of 5G core dedicated slices. We use OpenStack as VIM, Tacker as NFVO and VNFM and NYCU free5GC as the 5G core network functions.

We designed two 5GC Stage 3 systems: One consists of only UPF in the dedicated slices while the other consists of both SMF and UPF in the dedicated slices. We also provide a compared system which is based on free5GC Stage 2 with dedicated slices consisting of only UPF.

3.1. Free5GC Stage 3 with UPF Dedicated Slices

The architecture of free5GC Stage 3 with each dedicated slice consisting of only UPF is illustrated in **Figure 3(a)**. It uses three dedicated slices to handle three different types of test traffic. Each dedicated slice is connected to a specific DN (Data Network) server. Those UPFs in different dedicated slices share the same SMF in a common slice.

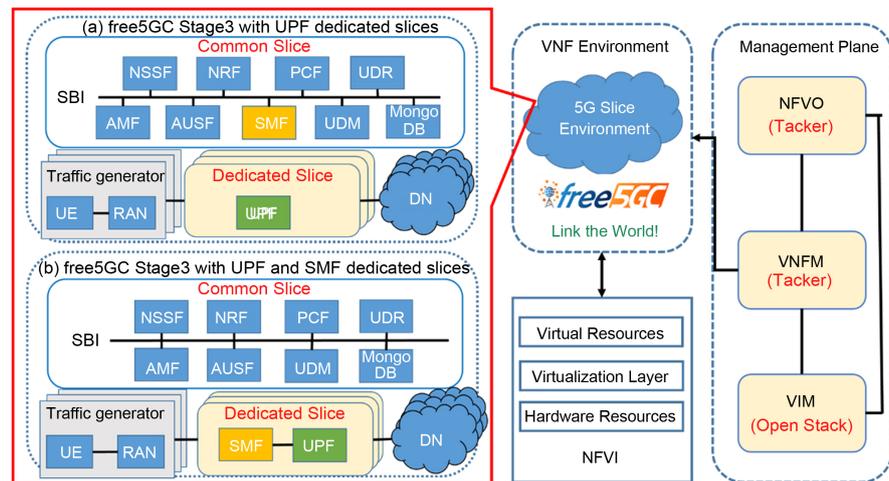


Figure 3. Free5GC with different configurations of dedicated slices.

A UPF can utilize all resources in a dedicated slice. The VNFs other than UPF reside in a common slice; these VNFs communicate with each other through Service Based Interface (SBI). There is a Mongo DB database used to store all system information such as valid users, SMF information, AMF information and policy rules.

In this system, we provide three traffic generators to simulate the packets that 5G UEs and RAN would send to 5GC. Each traffic generator would transmit two kinds of packets: UDP (User Datagram Protocol) and ICMP (Internet Control Message Protocol). Each traffic generator sends UDP packets at different rates: High, medium and low. All traffic generators send UDP packets and ICMP packets at the same time. We can estimate the throughput from the UDP traffic and calculate the response time from the ICMP packets. Each network slice routes the packets to a different Data Network (DN) server so we can validate its successful transmission through the chosen UPF.

The registration work flow is shown in **Figure 4**. First, the UE will connect to the RAN inside the traffic generator. Second, the traffic generator will send the NGAP (Next Generation Application Protocol) initial UE message to the AMF. This message carries the registration request and the UE information (include UE address IP, SST (Slice/Service Type) and SD (Slice Differentiator)). Third, the AMF requests UE authentication form the AUSF. If UE is a valid user, AUSF then accepts the request and sends the response to the AMF. After that, the AMF will send the UE information to the NSSF and then the NSSF will provide the available SMFs list to the AMF based on the information it received. Next, AMF will choose the proper SMF for the UE and create a smContext for SMF to set up a new session. Last, the SMF will choose an appropriate UPF and establish the PDU (Protocol Data Unit) session between the UE and the chosen UPF.

After registration is done, the UE can start to transmit the packets to its DN server. The work flow of transmission is shown in **Figure 5**. First, the UE sends UDP packets and ICMP packets to the UPF through its PDU session. Second,

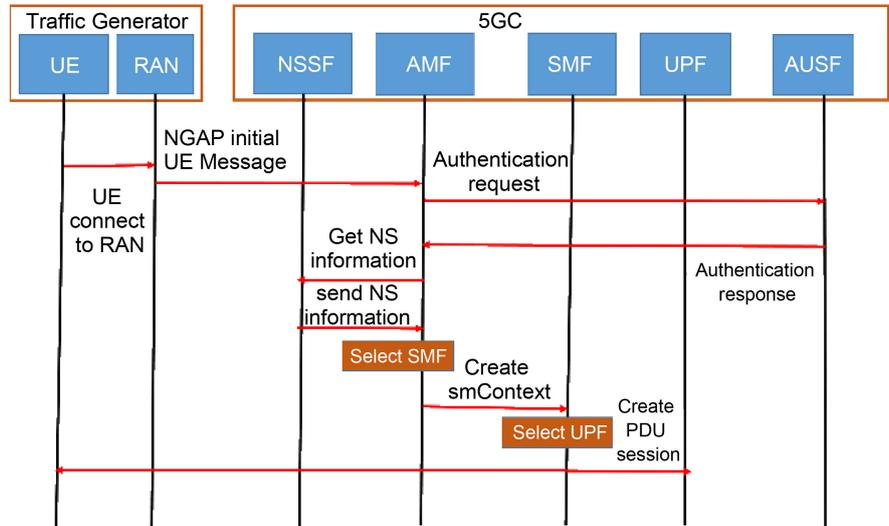


Figure 4. Registration workflow of free5GC Stage 3.

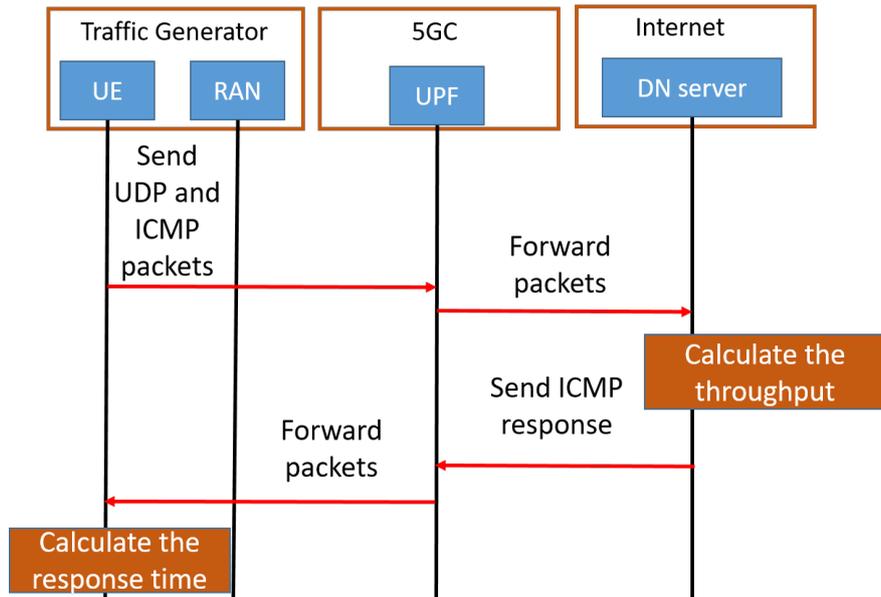


Figure 5. Transmission workflow.

the UPF will forward those packets to the specific DN. Third, the DN will calculate the throughput of UDP packets it received from the UPF. Fourth, when the DN receives the ICMP packets, it will send the ICMP response back to the UPF. Finally, the UPF will forward this ICMP response to the UE; this will allow the UE to calculate the ICMP response time.

3.2. Free5GC Stage 3 with SMF/UPF Dedicated Slices

The architecture of free5GC Stage 3 with each dedicated slice consisting of both SMF and UPF is illustrated in **Figure 3(b)**. It also uses three dedicated slices to handle different data rate requirements and each dedicated slice is connected to a specific DN server. All the VNFs are using the same resources as the previous

one. The difference from the previous one is that we move the SMF from the common slice to the dedicated slice. Each UPF is connected to an SMF in the same dedicated slice and both need to share the resources.

The traffic generators and DN servers are the same as those used in the previous architecture. The registration work flow is also similar to **Figure 4**; the only difference is that in the final step, SMF does not need to choose UPF, because the UPF is already assigned to each SMF. SMF only needs to establish the PDU session between the UE and the UPF. After registration is done, the work flow of transmission is also the same as that of the previous architecture as shown in **Figure 5**.

3.3. Free5GC Stage 2 with UPF Dedicated Slice

The architecture of free5GC Stage 2 with each dedicated slice consisting of only UPF is the same as the one shown in **Figure 3(a)**. The only difference is that the version of VNFs in this architecture is free5GC Stage 2. The VNFs in Stage 2 are less optimized than those in Stage 3.

The traffic generator and DN server are also the same as those in free5GC Stage 3. But the registration work flow is very different from Stage 3. This is because in free5GC Stage 2, NSSF is not utilized for the selection of a specifically dedicated slice to serve the requesting UE. It is assumed that UEs will send the traffic directly to an allocated slice as shown in **Figure 6**. First, the UE will connect to the RAN inside the traffic generator. Second, the traffic generator will send the registration request and also specify the UPF. Third, the AMF requests UE authentication from the AUSF. If the UE is a valid user, AUSF then accepts the request and sends the response to the AMF. Next, the AMF will pass the UE information to the SMF. Finally, the SMF finds the UPF specified by UE and establishes the PDU session between the UE and the UPF. On the other hand, the work flow of transmission for Stage 2 is the same as that for Stage 3 as shown in **Figure 5**.

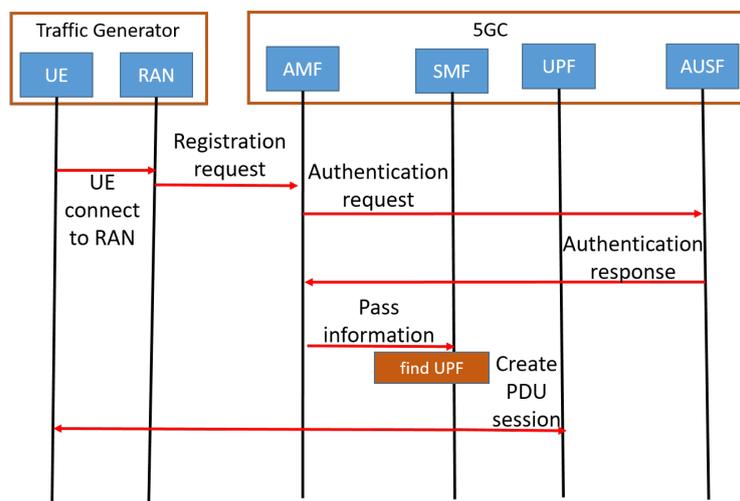


Figure 6. Registration workflow of free5GC Stage 2.

4. Implementation and Evaluation

In this section, we show the experimental setup followed by the evaluation results. In order to conduct a fair comparison, we adopt two different assumptions.

First, all the dedicated slices are allocated with the same amount of resources, *i.e.*, to use the same number of vCPU and the same amount of memory and storage.

Second, all three systems under evaluation are allocated with the same amount of resources. Below we show the results of testing three systems under both assumptions.

4.1. Environment Setup

We use two identical rack servers; one for OpenStack and another one for Tacker. **Table 1** shows the configurations of these two servers.

We follow the ETSI MANO framework discussed in Section II to build our slicing environment.

For the first assumption, the specifications of VNFs are separated into two parts: common slice and dedicated slice. For the common slice, since all the specifications of VNFs in the common slice are the same under different systems, we only use one entry to show them in **Table 2**. For the dedicated slices, we show different specifications of VNFs in **Table 3**. The configurations of three traffic generators and three DN servers are 2 vCPUs, 1 GB RAM and 10 GB disk with image Ubuntu 18.04.

For the second assumption, the specifications of VNFs are also separated into two parts: Common slice and dedicated slice. For the common slice, the resource

Table 1. Specification of implementation environment.

Entity	Operating System	Processor	RAM	HDD	Version
Tacker (NFVO, VNFM)	Ubuntu 18.04	Intel E5-2678V3 2.5 Ghz 10 Cores	128 GB	S4510 960 GB SSD	Stable Rocky
OpenStack (VIM)	Ubuntu 18.04	Intel E5-2678V3 2.5 Ghz 10 Cores	128 GB	S4510 960 GB SSD	Stable Train

RAM (Random Access Memory), HDD (Hard Disk Drive).

Table 2. Specification of VNF in common slice for first assumption.

Entity	Operating System	Processor	RAM	HDD	Version
Tacker (NFVO, VNFM)	Ubuntu 18.04	Intel E5-2678V3 2.5 Ghz 10 Cores	128 GB	S4510 960 GB SSD	Stable Rocky
OpenStack (VIM)	Ubuntu 18.04	Intel E5-2678V3 2.5 Ghz 10 Cores	128 GB	S4510 960 GB SSD	Stable Train

For the first assumption, SMF is included in this table. For the second assumption, SMF is not included in this table.

Table 3. Specification of VNF in dedicated slice for first assumption.

Entity	Operating System	vCPU	RAM	HDD
free5GC Stage 3 with UPF dedicated slices	Ubuntu 18.04	4	2 GB	20 GB
free5GC Stage 3 with SMF/UPF dedicated slices	Ubuntu 18.04	2	1 GB	10 GB
free5GC Stage 2 with UPF dedicated slices	Ubuntu 18.04	4	2 GB	20 GB

Dedicated slices include UPF and only free5GC stage3 with SMF/UPF dedicated slices has SMF in dedicated slice.

specifications of VNFs excluding SMF are the same as those in **Table 2**.

On the other hand, the resource specifications of SMFs under different configurations are shown in **Table 4**. Accordingly, the resource specifications of UPFs under different configurations are shown in **Table 5**.

Note that we make sure all the systems use the same number of vCPU and the same amount of memory and storage not only for SMF(s) but also for UPFs.

4.2. Experimental Results and Evaluation

In [22], the author proved and validated that the multiple network slicing system would have better throughput and response time compared to the one-slice-fits-all system. Their research showed that a multiple network slicing system provided better performance, but it didn't discuss how systems would be affected under different configurations. The experiment put all VNFs in a single network slice without separating them into a common slice and several dedicated slice. In this research, we go further to find out how the system will be affected if we move the SMF from the common slice to the dedicated slice by comparing three different system configurations. We evaluate the performance of these three systems by collecting their throughput, response time, CPU utilization and registration time under two different assumptions.

Figure 7 and **Figure 8** show the results of average throughputs of three different systems under three kinds of UDP packets data rates; low (80 Mbps), medium (200 Mbps) and high (400 Mbps) under two assumptions, respectively. Since our proposed systems using free5GC Stage 3 have been optimized, their throughputs are better than that of the compared system using free5GC Stage 2. Moreover, free5GC Stage 3 with the UPF dedicated slice provides higher throughput than free5GC Stage 3 with the SMF/UPF dedicated slice under the first assumption because the former has more vCPU resources than the latter. But if we give them the same vCPU resources as under the second assumption, their throughputs would be almost the same due to the same number of vCPUs.

The reason that our throughput cannot reach the expected goals of 80, 200 and 400 Mbps, respectively, for low, medium and high traffic is most likely because of the packet loss, not the performance of UPF. UPF can reach higher

Table 4. Specification of SMF for second assumption.

Entity	Operating System	vCPU	RAM/HDD	Numbers of SMF	location
free5GC Stage 3 with UPF dedicated slices	Ubuntu 18.04	6	3 GB/30GB	1	Common slice
free5GC Stage 3 with SMF/UPF dedicated slices	Ubuntu 18.04	2	1 GB/10GB	3	Dedicated slice
free5GC Stage 2 with UPF dedicated slices	Ubuntu 18.04	6	3 GB/30GB	1	Common slice

Table 5. Specification of UPF for second assumption.

Entity	Operating System	vCPU	RAM	HDD
UPF in Dedicated slice	Stage 3: Ubuntu 18.04 Stage 2: Debian 10	2	1 GB	10 GB

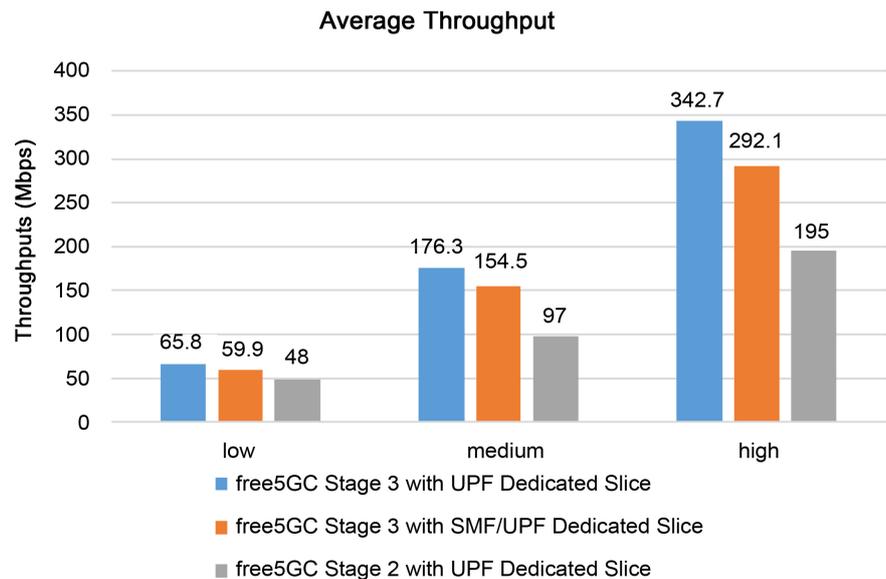


Figure 7. Average throughput under first assumption.

throughput when traffic generator sends higher data rate.

Figure 9 and **Figure 10** show the average response times of three different systems under two assumptions. Because of the design of our proposed system is more complicated than that of the compared system, the average response times become longer. Under the second assumption both systems have similar response times. But under the first assumption the response time of the free5GC Stage 3 with the UPF dedicated slice is shorter because the UPF has more vCPU resources.

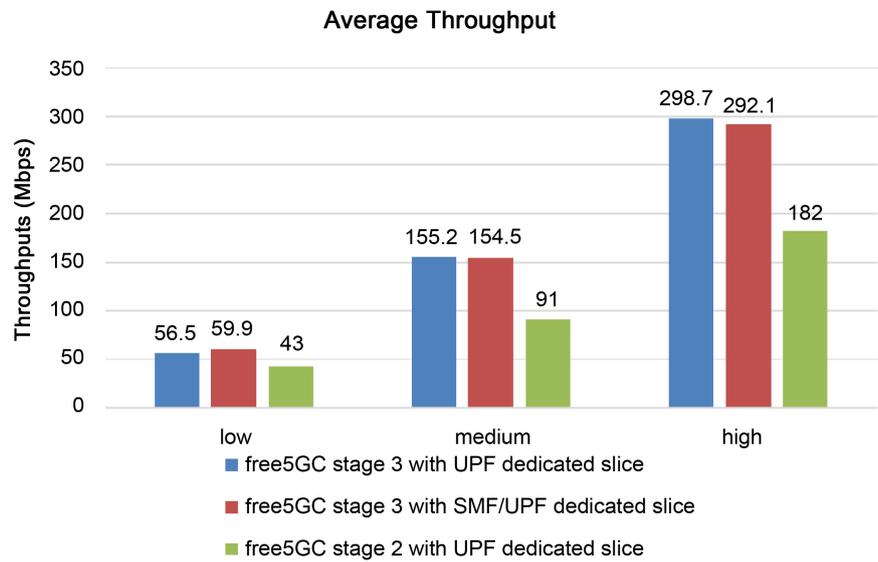


Figure 8. Average throughput under second assumption.

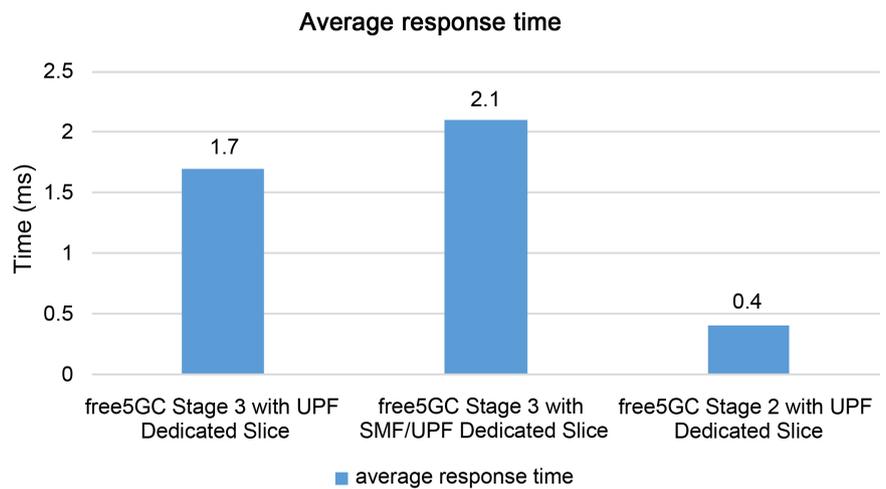


Figure 9. Average response time under first assumption.

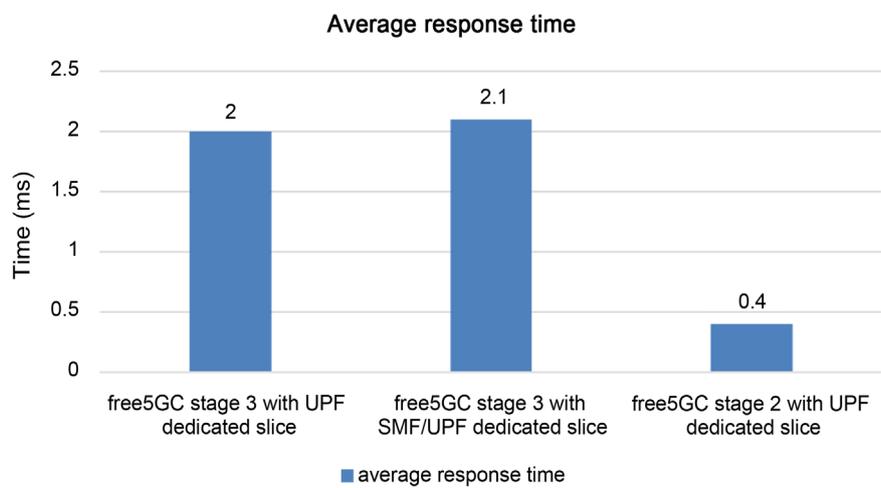


Figure 10. Average response time under second assumption.

Figure 11 and Figure 12 show the average CPU utilizations of two free5GC Stage 3 systems under three kinds of data rates. Under the second assumption, there is no difference between two free5GC Stage 3 systems on CPU utilization. However, under the first assumption the CPU utilizations are almost the same even though the numbers of vCPU used by UPFs are not the same. But in both assumptions, if we send more packets from traffic generators, the CPU utilizations become higher. This is because the CPU utilization depends only on the number of packet transmission. On the other hand, the CPU utilization of free5GC Stage 3 with UPF dedicated slices is a little higher than that of free5GC Stage 3 with SMF/UPF dedicated slices because using more vCPU resources has caused more race conditions among CPUs.

Figure 13 and Figure 14 show the average registration time in our proposed systems. In the free5GC Stage 3 with SMF/UPF dedicated slices, there is no need

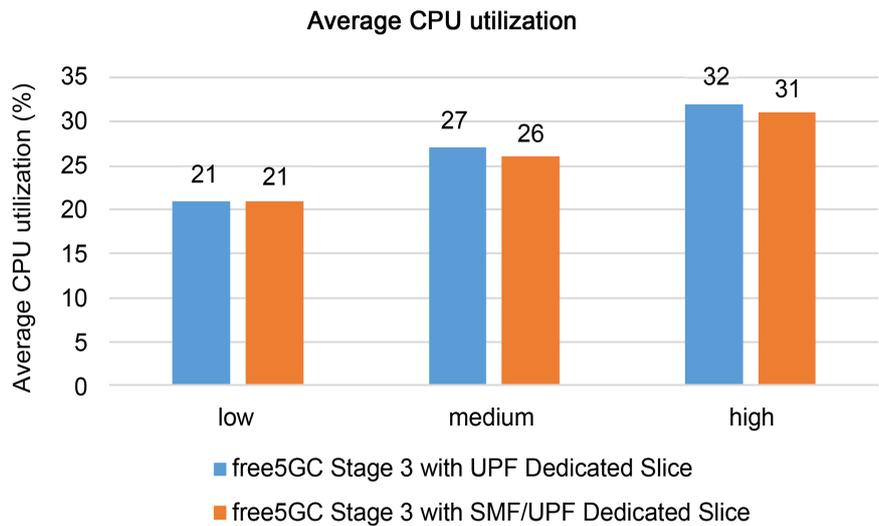


Figure 11. Average response time under second assumption.

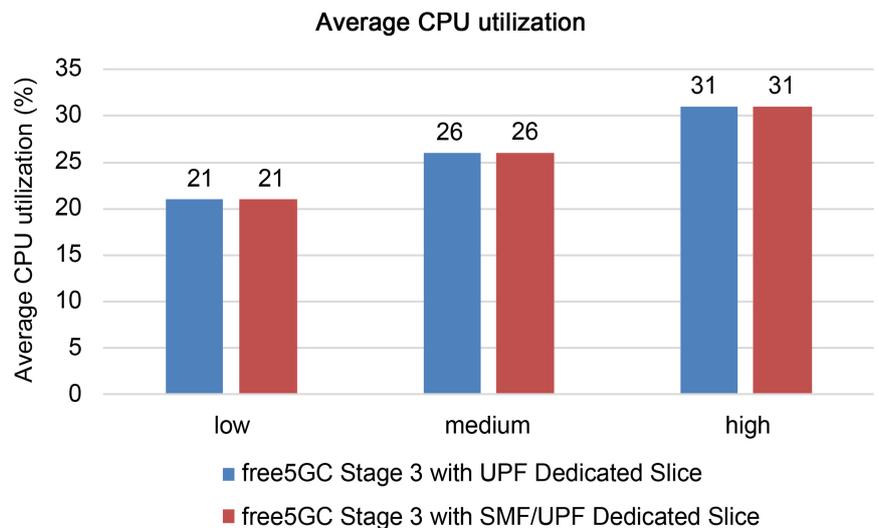


Figure 12. Average CPU utilization under first assumption.

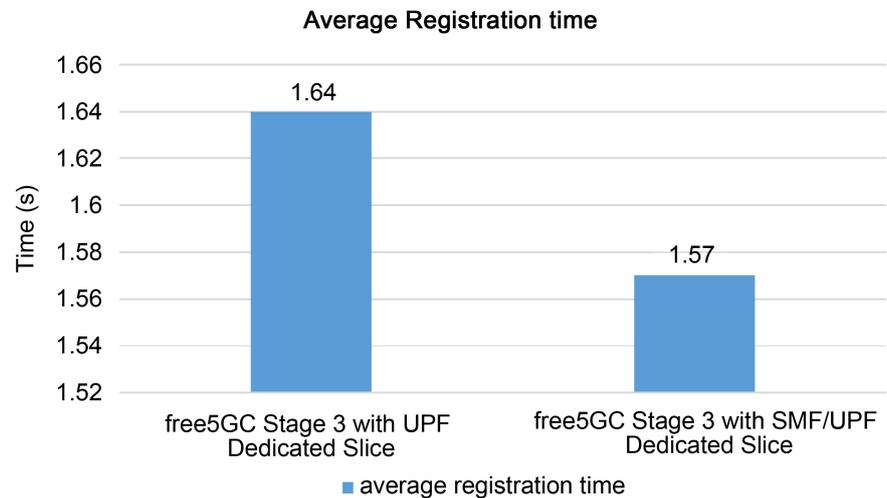


Figure 13. Average registration time under first assumption.

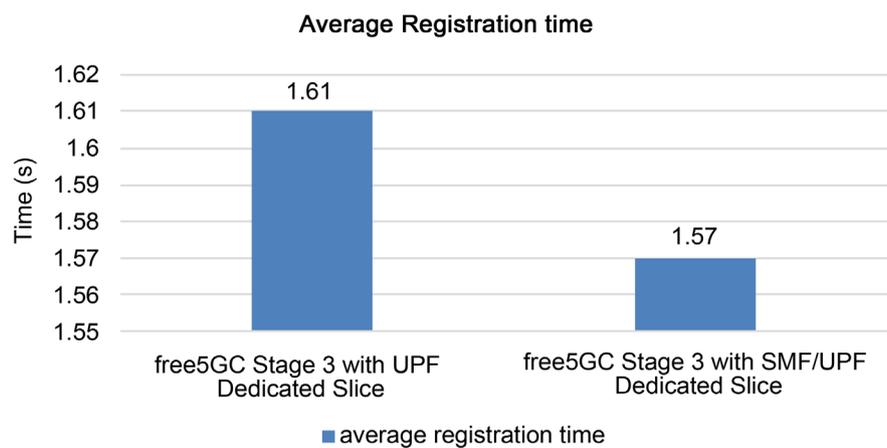


Figure 14. Average registration time under second assumption.

for the SMF to choose the UPF, so the registration time is lower under both assumptions. Although under the second assumption we provided a more powerful SMF in free5GC stage 3 with UPF dedicated slices, it still took longer time during registration. This is because it needs to spend extra time for the SMF to choose the UPF.

5. Conclusions & Future Work

In this paper, some open source projects are leveraged such as free5GC, OpenStack, Tacker to experiment with the deployment of dedicated slices under different architectural configurations. The performances of three system architectures are also compared in terms of throughput, response time, CPU utilization and registration time.

The results conclude the following: Moving SMF from the common slice to the dedicated slice under the first assumption would shorten the registration time but worsen the performance of UPF since the resources allocated to UPF are less than before. But under the second assumption, only the registration time

will be affected; the performance of UPF will not be affected. This is because when the transmission starts, the functions in control plane are no longer participating in the operations.

Thus if a large number of connections is required in a short period of time, moving SMF from the common slice to the dedicated slice would be a better choice because this system has the lower registration time; it can handle a large number of registrations more efficiently.

Also, if the user wants to have better throughput and shorter response time under the first assumption, it is recommended to keep SMF in the common slice so that the UPF can be allocated with more resources for better performance.

Though not conducting experiments yet, it is predicted that if moving more control plane VNFs such as AMF, NRF to the dedicated slices, the registration time will become even shorter. This is because that all the paths are predefined; there is no need for any selection.

In the future, more experiments could be conducted. First, experiments with different resource allocations to dedicated slices could be done, such as decreasing the resources allocated to SMF but increasing the resources allocated to UPF in the dedicated slice. Second, we can experiment with more configurations of dedicated slices and identify more use cases best suitable for each different configuration. Finally, incorporating scalability capability into dedicated slices could also be done, so they can scale in/out or up/down automatically.

Acknowledgements

This work was financially supported by (1) the Center for Open Intelligent Connectivity from The Featured Areas Research. Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan and (2) the Ministry of Science and Technology (MOST) of Taiwan under MOST 109-2221-E-009-083.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Napolitano, A., Giorgetti, A., Kondepu, K., Valcarengi, L. and Castoldi, P. (2018) Network Slicing: An Overview. 2018 *IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, Palermo, Italy, 10-13 September 2018, 1-4. <https://doi.org/10.1109/RTSI.2018.8548449>
- [2] Gomes, R.L., Urbano, A., da Ponte, F.R.P., Bittencourt, L.F. and Madeira, E.R.M. (2018) Slicing of Network Resources in Future Internet Service Providers Based on Daytime. 2018 *IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, 25-28 June 2018, 1-4. <https://doi.org/10.1109/ISCC.2018.8538725>
- [3] Zhou, X., Li, R., Chen, T. and Zhang, H. (2016) Network Slicing as a Service: Enabling Enterprises' Own Software-Defined Cellular Networks. *IEEE Communications Magazine*, **54**, 146-153. <https://doi.org/10.1109/MCOM.2016.7509393>

- [4] Free5GC.org (2019) <https://www.free5gc.org/>
- [5] Ma, L., Wen, X., Wang, L., Lu, Z. and Knopp, R. (2018) An SDN/NFV Based Framework for Management and Deployment of Service Based 5G Core Network. *China Communications*, **15**, 86-98. <https://doi.org/10.1109/CC.2018.8485472>
- [6] Atoui, W.S., Assy, N., Gaaloul, W. and Ben Yahia, I.G. (2020) Learning a Configurable Deployment Descriptors Model in NFV. 2020 *IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, 20-24 April 2020, 1-9. <https://doi.org/10.1109/NOMS47738.2020.9110328>
- [7] Chen, J., Cao, H. and Yang, L. (2019) NFV MANO Based Network Slicing Framework Description. 2019 *IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW)*, Taiwan, 20-22 May 2019, 1-2. <https://doi.org/10.1109/ICCE-TW46550.2019.8991887>
- [8] OpenStack (2020) <https://www.openstack.org/>
- [9] OpenStack (2020) OpenStack Tacker Project. <https://docs.openstack.org/tacker/latest/>
- [10] Mandal, P. and Jain, R. (2019) Comparison of OpenStack Orchestration Procedures. 2019 *International Conference on Smart Applications, Communications and Networking (SmartNets)*, Sharm El Sheik, Egypt, 17-19 December 2019, 1-4. <https://doi.org/10.1109/SmartNets48225.2019.9069772>
- [11] Mamushiane, L., Lysko, A.A., Mukute, T., Mwangama, J. and Toit, Z.D. (2019) Overview of 9 Open-Source Resource Orchestrating ETSI MANO Compliant Implementations: A Brief Survey. 2019 *IEEE Wireless Africa Conference (WAC)*, Pretoria, South Africa, 18-20 August 2019, 1-7. <https://doi.org/10.1109/AFRICA.2019.8843421>
- [12] Zhou, X., Li, R., Chen, T. and Zhang, H. (2016) Network Slicing as a Service: Enabling Enterprises' Own Software-Defined Cellular Networks. *IEEE Communications Magazine*, **54**, 146-153. <https://doi.org/10.1109/MCOM.2016.7509393>
- [13] Sattar, D. and Matrawy, A. (2019) Optimal Slice Allocation in 5G Core Networks. *IEEE Networking Letters*, **1**, 48-51. <https://doi.org/10.1109/LNET.2019.2908351>
- [14] Li, X., *et al.* (2017) Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Computing*, **21**, 20-27. <https://doi.org/10.1109/MIC.2017.3481355>
- [15] Kaloxylos, M. (2018) A Survey and an Analysis of Network Slicing in 5G Networks. *IEEE Communications Standards Magazine*, **2**, 60-65. <https://doi.org/10.1109/MCOMSTD.2018.1700072>
- [16] Kozłowski, K., Kukliński, S. and Tomaszewski, L. (2018) Open Issues in Network Slicing. 2018 *9th International Conference on the Network of the Future (NOF)*, Poznan, Poland, 19-21 November 2018, 25-30. <https://doi.org/10.1109/NOF.2018.8598130>
- [17] Kukliński, S. and Tomaszewski, L. (2019) Key Performance Indicators for 5G Network Slicing. 2019 *IEEE Conference on Network Softwarization (NetSoft)*, Paris, France, 24-28 June 2019, 464-471. <https://doi.org/10.1109/NETSOFT.2019.8806692>
- [18] Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A. and Flinck, H. (2018) Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Communications Surveys & Tutorials*, **20**, 2429-2453. <https://doi.org/10.1109/COMST.2018.2815638>
- [19] Diaz Rivera, J.J., Khan, T.A., Mehmood, A. and Song, W. (2019) Network Slice Selection Function for Data Plane Slicing in a Mobile Network. 2019 *20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Matsue, Japan,

- 18-20 September 2019, 1-4. <https://doi.org/10.23919/APNOMS.2019.8893084>
- [20] Wei, H., Zhang, Z. and Fan, B. (2017) Network slice access selection scheme in 5G. 2017 *IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 15-17 December 2017, 352-356. <https://doi.org/10.1109/ITNEC.2017.8284751>
- [21] Choyi, V.K., Abdel-Hamid, A., Shah, Y., Ferdi, S. and Brusilovsky, A. (2016) Network Slice Selection, Assignment and Routing within 5G Networks. 2016 *IEEE Conference on Standards for Communications and Networking (CSCN)*, Berlin, Germany, 31 October-2 November 2016, 1-7. <https://doi.org/10.1109/CSCN.2016.7784887>
- [22] Liao, C.W., Lin, F.J. and Sato, Y. (2020) Evaluating NFV-enabled Network Slicing for 5G Core. 2020 *21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Daegu, Korea (South), 22-25 September 2020, 401-404. <https://doi.org/10.23919/APNOMS50412.2020.9236978>