Computer Communications 149 (2020) 252–258

Contents lists available at ScienceDirect

Computer Communications

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

Towards 5G network slicing for vehicular ad-hoc networks: An end-to-end approach

Muhammad Afaq [a], Javed Iqbal [a,*], Talha Ahmed [b], Ihtesham Ul Islam [a], Murad Khan [a], Muhammad Sohail Khan [c]

[a] Sarhad University of Science and IT, Peshawar, 25000, Pakistan
[b] Jeju National University, South Korea
[c] Department of Computer Software Engineering, University of Engineering and Technology, Mardan, Pakistan

ARTICLE INFO

Keywords:
Network slicing
5G
M-CORD
LTE
NSSF
VANETs

ABSTRACT

The upcoming 5G networks not only have to support increasing data rates but also must provide a common infrastructure on which new services with vastly different network QoS requirements with lower delay are delivered. More precisely, applications for VANETs, that are mainly oriented to safety issues and entertainment (e.g. video streaming and video-on-demand, web browsing) are increasing. Most of these applications have strict latency constraints of the order of few milliseconds, and very high reliability requirements. To address such needs, a 5G platform needs the ability to dynamically create programmable virtual networks and differentiated traffic treatment utilizing solutions such as network slicing. To this end, in this paper, we propose a programmable and dynamic end-to-end slicing mechanism in an M-CORD based LTE network. One of the key features of M-CORD that the proposed network slicing mechanism utilizes is the virtualized EPC that enables customization and modification. M-CORD provides necessary functionality to program slice definitions, where the proposed mechanism fully follows its software-defined approach. Furthermore, we demonstrate how end devices placed in different slices can be allocated with different QoS treatments from the network operator based on end-user type. The results show that the proposed network slicing mechanism selects appropriate slices and allocates resources to users specific to their needs and service type.

1. Introduction

The evolving technology trends entail a radical innovation in the next generations of mobile networks. Previous generations of mobile networks enabled voice, video, data, and other life-changing services. In comparison, the recent transition toward the 5th generation network will enable a vast domain of new use cases by achieving a scalable, flexible and optimized network [1]. On the other hand, previous network architectures lack the ability to meet the specific needs that users, operators, applications, dynamic use-cases require from the network due to their embedded nature. Consequently, they allocate identical resources to all type of network traffic regardless of their requirement [2]. Lack of capabilities in existing framework forced researchers to design novel architectures that should meet dynamic user demands [3]. For achieving the vision of 5G a combination of SDN (Software Defined Networking) and NFV (Network Function Virtualization) serves as the foundation for technological improvements. Furthermore, they enable network slicing hence introducing scalability, flexibility, and optimization of resource usage in network architectures. More importantly, NFV and SDN enable programmability on the user plane and control plane respectively providing a complete independence from specific hardware restricted networking. One can program specified functionality for the user-plane and control plane function to achieve required results i.e. a specific function with specific QoS settings can be introduced to cater advanced services like Vehicular network and IoT (Internet of Thing). Moreover, the core of the enhancements in mobile network lies in resource allocation optimization. For this reason, network slicing divides the physical network into multiple logical networks [4]. Ultimately, it is an esteem requirement to develop novel network slicing architecture.

The platform that combines SDN and NFV functionality under one system is foundation for the development of the next-generation mobile networks [5]. Pursuing the fact, M-CORD (Mobile-Central Office Re-architected as Datacenter) is an open reference solution developed by Open Networking (ON) lab [6]. It stands on the pillar of SDN and NFV with an aim to transform telco office into a cloud Data Center. In fact, CORD [7] grants independence from embedded hardware by converting legacy NF (network functions) with VNF's (virtual network

* corresponding author
E-mail addresses: afaq.csit@suit.edu.pk (M. Afaq), javed.ee@suit.edu.pk (J. Iqbal), talhajadun@gmail.com (T. Ahmed), ihtesham.csit@suit.edu.pk (I.U. Islam), murad.csit@suit.edu.pk (M. Khan), sohail.khan@uetmardan.edu.pk (M.S. Khan).

https://doi.org/10.1016/j.comcom.2019.10.018
Received 14 July 2019; Received in revised form 30 September 2019; Accepted 18 October 2019
Available online 24 October 2019
0140-3664/

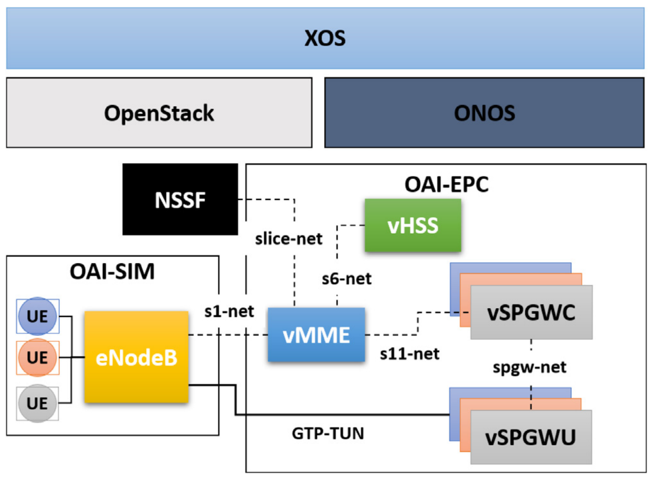

Fig. 1. Depiction of the overall architecture. NSSF (Network Slice Selection Function) selects a path for connection to the DN (Data Network) based on the information sent by the UE. The open-source eNodeB and EPC (evolved packet core) complete the end-to-end network slicing framework. The overall system is developed on top of M-CORD, a matured platform for slicing.

function) also known as service in CORD context that exist on the commodity servers. The CORD control platform consists of a combination of XOS (Service Orchestrator) that provides XAAS (Everything as a Service) i.e. every NF can be developed as a service consisting of multiple service instance each with different QoS subscription in accordance to SLA (service level agreements). ONOS (Open Networking Operating System) that equips CORD with all the networking capabilities and OpenStack that provides Infrastructure as a Service for CORD [8].

3GPP (3rd Generation Partnership Project) is a standardization body that provides basis for the development of standard network infrastructures. While considering the standards, this manuscript includes firstly, OAI-SIM (Open Air Interface Simulator) [9] which simulates eNodeB and can support up to twenty UE (User Equipment). Secondly, NSSF (Network Slice Selection Function) which is used to select different slices for each type of user request. Thirdly, OAI-EPC (Open AIR Interface Evolved Packet Core) which in turn includes vMME (virtual Mobility Management entity), vHSS (Virtual Home Subscriber Server Service), vSPGWC (virtual Serving and Provider-Data-Network Gateway Control Plane), vSPGWU (virtual Serving and Provider-Data-Network Gateway User Plane). All of the above NF's are well defined in the standard documents and upcoming section will detail about their functionality and deployment under M-CORD.

This paper focuses on the fundamental aspect for the development of the next-generation mobile network for catering multiple applications including eMBB, V2X, IoV [10,11], and IoT. To this purpose, network slicing is implemented by following the reference of the 5G architecture. This work includes a special network function called the NSSF (Network Slice Selection Function). It is capable for selecting a path for connection to the DN (Data Network) based on the information sent by the UE; the functionality of this component will be explained in Section 3. In addition to that, we introduce the open source eNodeB and EPC (evolved packet core) to complete the end to end network slicing framework. The overall system is developed on top of a matured platform for slicing namely M-CORD.

The rest of the paper is organized as follows. Section 2 presents related work, Section 3 briefs system overview, Section 4 explains system components, Section 5 briefs network slicing on management plain and results achieved using the purposed architecture, and Section 6 concludes the paper.

2. Related work

In [12], authors have proposed a network slicing framework together with mechanisms and relevant building blocks to handle V2X applications by edge-dominated 5G network infrastructures and flexibly orchestrating multi-access, especially with reference to roaming scenarios. Experiments using the Mininet emulator show the potential and feasible benefit of the proposed framework for cooperative driving use cases.

In [13], the design of 5G network slices tailored for vehicle-to-everything services, which involve data exchange between vehicles, with any communicating entity and infrastructure for enhanced transport flexibility, comfort, and safety on the road. To support different vehicle-to-everything use cases, the suggested slicing solutions involve configuration of the vehicular end-device functionality, as well as the partitions of the radio access network and core network resources. The authors elaborate on the technological enablers and option, challenges and concerns for the deployment of 5G slices for multitenant vehicle-to-everything services.

In [14], the authors model a highway scenario with vehicles having heterogeneous traffic requirements. The autonomous driving slice (safety messages)and the infotainment slice (video stream) are the two logical slices created on a common infrastructure. We formulated a network clustering and slicing algorithm to partition the vehicles into different clusters and allocate slice leaders (SLs) to each cluster. SLs serve its clustered vehicles with high-quality V2V links and forwards safety information with low latency. On the other hand, road side unit provides infotainment service using high-quality V2I links. An extensive Long Term Evolution Advanced system-level simulator with enhancement of cellular V2X standard is used to evaluate the performance of the proposed method, in which it is shown that the proposed network slicing technique achieves low latency and high-reliability communication.

In contrast to the aforementioned approaches, our proposed approach implements an end-to-end network slicing on an M-CORD-based platform. More precisely, a network function, NSSF has been implemented for selecting the appropriate end-to-end virtual network. This selection is made on the basis of the information received from the UE. Furthermore, open-source eNodeB and EPC have been implemented in the platform to establish several end-to-end network slices for different services.

3. System overview

The system that allows us to showcase our Slice Selection scenario consists of the following components: Orchestrator: Fulfilled by XOS (Everything as a Service) and is the most essential component of the CORD platform. By treating each component of a mobile network as a Service, XOS abstracts the interactions that allow the creation of multiple VNF's namely veNodeB, vMME, vNSSF, vHSS, vSPGWC and vSPGWU in collaboration with a network controller and a cloud manager as shown in Fig. 1.

Network Controller: ONOS is the default network controller used in our scenario. As the CORD platform is being developed by the ONF (Open Network Foundation). ONOS is also developed by ONF as an enabler for SDN, the use of a network controller is essential, in this case. ONOS provides the means for interconnecting the VNF that are going to be instantiated in our scenario and also the data network access required for an E2E connection in a mobile scenario. This is achieved by two ONOS applications called vRouter and VTN. To elaborate, vRouter configures routing protocols for the network topology. On the other hand, VTN is used for interconnecting VNF's by allocating virtual network namely s1net, N-net etc as shown in Fig. 1.

Cloud Manager: Virtual Machine and Network Instantiation is managed directly by OpenStack [15] and overview of procedure is shown in Fig. 2. Also, the VNF that represent the Mobile Network are an implementation of the Open Source Project — Open Air Interface for both the EPC and the veNodeB as illustrated in Fig. 1.

Our main contribution to this scenario, is the inclusion of the vNSSF as part of the VNF of mobile network, its functionality is going to be detailed in Section 3.

Fig. 2 shows step-wise working of management plane, Firstly XOS feeds TOSCA configuration file which contains the system architecture and service configurations. Secondly, XOS core compiles the configurations to build models. As a third step it provides them to XOS Database. After this in the fourth step the database stores configurations as VNF models. In the fifth step, the OpenStack synchronizer that keeps track of every update in the XOS DB detects the configuration changes. Finally, OpenStack as resource provider allocates VM and VN (virtual Networks) for the required TOSCA configurations. Where VM are provided by the OpenStack solely while VN are provided by OpenStack and configured using VTN application of ONOS. Basically, the TOSCA is provided to M-CORD by the network administrator and it contains the configuration of VNF services under CORD. For each service the administrator defines multiple instances each with specific configurations. The specific configurations define the QoS and other SLA capabilities so that these instances can be allocated on runtime to certain users. So once TOSCA service graph is fed to M-CORD it creates different VNF's with multiple instances and defines networking between them using synchronizers. Synchronizers include OpenStack VNF synchronizer and ONOS application synchronizers. Hence, M-CORD provides an open platform where one can determine its own architecture for this work we designed our own architecture to cater multiple services that require different capabilities. Fig. 3 shows a similar architecture where a service graph with OAI-SIM and OAI-EPC component has been deployed as CORD services. This architecture additionally introduces vNSSF which enhances the slicing concepts and enables 5G network to select multiple slices on runtime in accordance with UE service request. The multiple instances of vSPGWC and vSPGWU are introduced and each instance can have different QoS capabilities to serve specified services which can include Va-net slice or other services that 5G offer. Fig. 3 explains that different synchronizers are responsible for orchestrating each type of service where EPC and vNSSF are synchronized using EPC-Sync and NSSF-Sync. Also, the VTN-synch orchestrates different network that connects different network function and is depicted as Nssf-net, s6 and so on. However, we can see that each UE (user equipment) has unique data plane path and have different control plane function in EPC. So, the conclusion is multiple network slices can be defined with different QoS under one single CORD platform.

4. System components

This section details the system component which is developed as part of 5G architecture namely NSSF. In addition, it presents the details of open-source components, OAI-eNB and OAI-EPC, that are used and modified for the implementation of different scenarios in this paper.

4.1. NSSF

Since, network slicing is one of the major characteristics of a 5G network, the existence of an NSSF becomes vital. More precisely, it is responsible for the selection of appropriate network slice on the basis of parameters sent be a UE. The role of vNSSF along with other network functions in 5G architecture is depicted in Fig. 4. vNSSF selects the Network Slice Instance (NSI), determines the allowed Network Slice Selection Assistance Information (NSSAI) and Access and Mobility Management Function (AMF) set to serve the UE (see Fig. 4).

As mentioned before, each VNF is represented by a service, this service has direct communication with XOS, and inside this service there is synchronizer code that can be used to provide information that otherwise will not be accessible between VNF (As they are running as

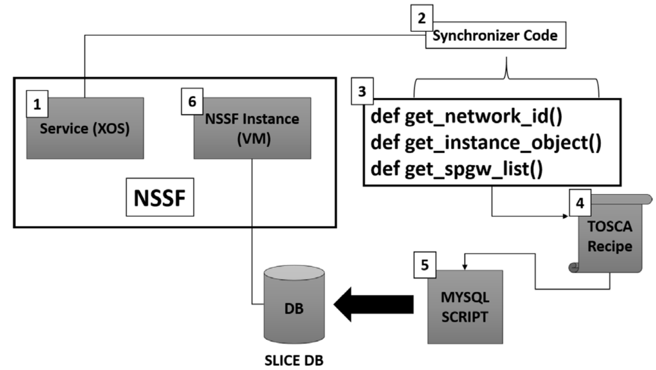

Fig. 5. Representation of the process of how the vNSSF gets to know the slice information of the underlying mobile network. In XOS DB the information of all the VNF instantiated is available and the vNSSF synchronizer keeps track of the DB updates on every change it updates its information. As CORD platform provides the scalability and service configuration update features, more instances can bee added to network.

isolated VMs). During VNF instance creation, a TOSCA recipe is fed to the vNSSF. The recipe holds the information of how many vSPGW have been instantiated for the mobile network. Furthermore, because of the native synchronizer functionality of the services that run in XOS, this information is updated every time whenever there is an underlying change in each of the VNF Instances. Practically the XOS Orchestrator provides the means for the NSSF to know the exact number of slices that exist for selection. Fig. 5 represents the process of how the vNSSF gets to know the slice information of the underlying mobile network. Basically, in XOS DB the information of all the VNF instantiated is available and the vNSSF synchronizer keeps track of the DB updates on every change it updates its information. Hence every new entry will automatically be synchronized. Furthermore, as CORD platform provides the scalability and service configuration update features that is we can add more instances to our network at any time and through NSSF synchronizer procedure those updates will be synchronized automatically and NSSf at every request will be providing the updated selection of the resources.

4.2. OAI SIM

The OAI project that contains UE and the veNodeB implementations is called openairinterface. The project is freely distributed by the OpenAirInterface Software Alliance (OSA) and allows contributions from members holding patents on key procedures used in the standards. The software provides its own implementation for veNodeB and UE emulation/simulation called OpenAirInterface System Emulation (OAISIM) as shown in Fig. 6.

OAISIM implementation is selected for the implementation of access network in this work. It provides simulation with full physical and synthetic radio channels. It also offers simulation of up to twenty UEs that can act simultaneously. In this work, tests have been performed by using two UEs simultaneously. In this work, tests have been performed by using two UEs simultaneously. We considered each UE requests a different service so we allocated different slices to each of the resources. The UEs are simulated so we tested the slice capabilities that how each slice sticks to the configurations we provided through TOSCA and we showed in our results how different SLA can be allocated and managed in 5G networks.

4.3. OAI-EPC

The OSA's EPC software is known as OpenairCN. It contains implementation of vMME, vHSS, vSPGW-C and vSPGW-U collectively serving

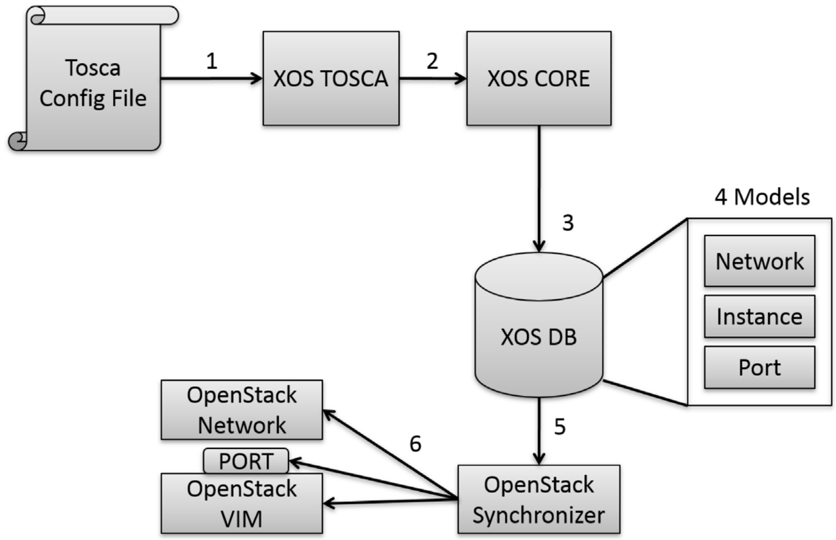

Fig. 2. Illustration of working of management plane. XOS feeds TOSCA the configuration file which contains the system architecture and service configurations, as well as compiles the configurations to build models. It provides them to XOS Database. The database stores configurations as VNF models. The OpenStack synchronizer that keeps track of every update in the XOS DB detects the configuration changes. OpenStack as resource provider allocates VM and VN (virtual Networks) for the required TOSCA configurations.

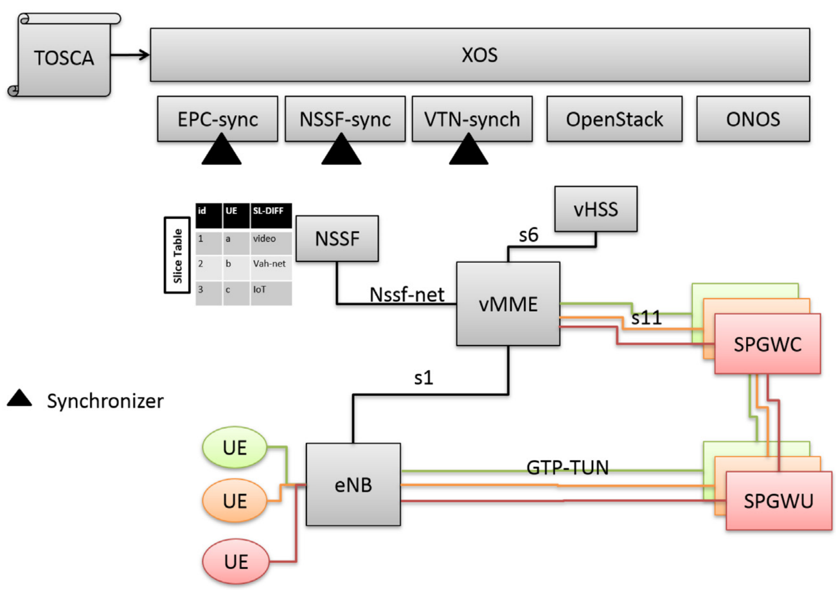

Fig. 3. Depiction of architecture where a service graph with OAI-SIM and OAI-EPC component have been deployed as CORD services. This architecture additionally introduces vNSSF which enhances the slicing concepts and enables 5G network to select multiple slices on runtime in accordance with UE service request. The multiple instances of vSPGWC and vSPGWU are introduced. Different synchronizers are responsible for orchestrating each type of service where EPC and NSSF are synchronized using EPC-Sync and NSSF-Sync.

as core network. It is distributed under an Apache V2.0 license, in order to ease with integration within an OpenStack environment.

The vMME connects the access network with other core network components. It handles mobility-related issues by keeping track of the user movements using the information provided by the vBBU. It performs required actions for updating connection between core and access networks on S1u interface using the vSPGW-C, as shown in Fig. 7.

The vHSS is responsible for storing the subscription information of the network. It communicates with the vMME to validate the UE information that is sent from the veNB in order to compare it with information stored in a database.

The vSPGW-C manages sessions between access, core and data network. It also defines end-to-end protocols for user plane traffic. Moreover, it controls vSPGW-U using interface S5, and collects information of access network from vMME using S11 interface.

The vSPGW-U is responsible for data communication between access and data network. It generates S1-U tunnel and SGI tunnel for every slice. This is represented by the creation of a GTP-Tunnel between the UE and the vSPGW-U upon PDU session establishment [16].

5. Results and discussion

The testbed deployment is based on M-CORD version 4.1 as a cord in a box (CiaB) which allows for a virtualized environment that can exist in a single physical server. For setting up the environment, three VM images are created that can be used by Openstack. The EPC image is obtained from the oai-scenario repo of the developer aweimeow. The veNB and vNSSF images are created as part of this work.

OAISIM is the actor that triggers the mobile network test. It is configured with two UEs that have different IMSI. The vMME gets

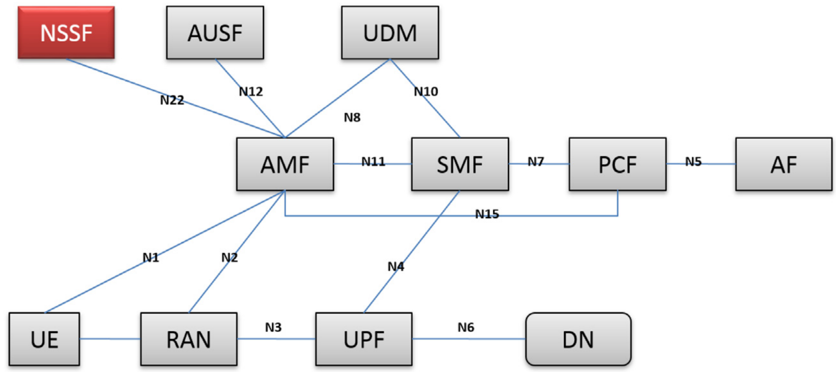

Fig. 4. Illustration of the role of NSSF along with other network functions in 5G architecture. NSSF selects the Network Slice Instance (NSI), determines the allowed Network Slice Selection Assistance Information (NSSAI) and Access and Mobility Management Function (AMF) set to serve the UE.

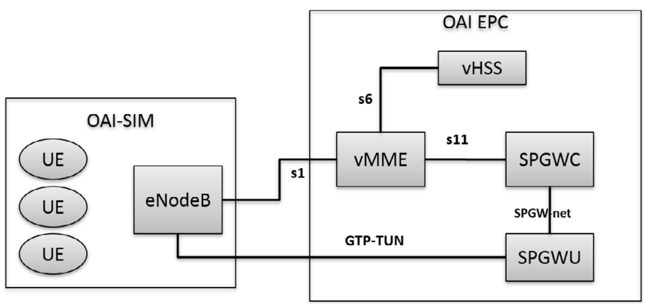

Fig. 6. The OAI project that contains UE and the eNodeB implementations is called openairinterface. OAISIM implements access network, and provides simulation with full physical and synthetic radio channels.

this information from the veNB and by communicating with the vHSS, verifies its IMSI. Once the UE is registered into the network, the original OAI implementation of the vMME would select a vSPGW on the basis of the TAC and TAI information that the eNB sent during network registration.

The vNSSF will have full control of the slice selection. It is in this module where specific vSPGW are related to defined types of services. It depends on network operators on how these types of services are going to be categorized. As explained before, two types of services have been defined: WEB and VIDEO, in turn two vSPGW were assigned to each of these services. In our scenario the vNSSF will check for the IMSI of the UE and if the last digit matches a defined service, it will reply to the vMME with the id of the SPGW (IP address) from where the process will continue to reach the data session establishment.

As long as Types of Services are well defined, there is the possibility to create as many slices as the capacity of the system allows it. Each vSPGW instance can be configured with simple (or more complex depending on the scenario) QoS policies that can affect the Latency and Bandwidth of the connection.

OAISIM is not a reliable medium for performing performance tests. Although it theoretically supports rates of 20 Mb/s to 30 Mb/s thanks to the implementation of the OAI veNB, the emulator only has a throughput of around 1 Mb/s on the UE that are being emulated, regardless of the slice that it has been connected. For testing purposes, we use the iperf utility available in the ubuntu distribution on the virtual machine where OAISIM is running. In order to have a reliable test, iperf connects to a server through the internet. i.e. the SPGW-U provides Internet Access to another physical server having a public IP Address. A full E2E connection test of a UE connected to the default slice shows the behavior as shown in Fig. 8.

It is important to note, that a huge load of processing data is being carried by OAISIM, and most of the tests do not complete, or show unreliable values. The data shown in Fig. 8 is a compilation of what we consider the most reliable data that we could achieve with a connection from the UE to the Data Network.

Due to the bottleneck that OAISIM was causing for the performance test, iperf is directly run on the vSPGW Instances that were configured to serve specific type of services (WEB and VIDEO). The vSPGW that acts as WEB slice is configured to cap the connection at 10 Mb/s, on the other hand the vSPGW that serves the VIDEO slice is uncapped and its throughput depends on the link capabilities. In our lab we have a 100 Mb/s Internet connection hence results are lagging the actual performance consideration.

The test performed on the WEB — vSPGW shows that the connection is more reliable compared to the OAISIM tests. Although the bandwidth cap is set to 10 MB/s for this slice, the link sometimes went over the cap or under it, which can be attributed to link as well as server performance issues due to running a considerable amount of VM Instances on the single physical server as shown in Fig. 8. On the other hand, the tests performed on the VIDEO — vSPGW, show more

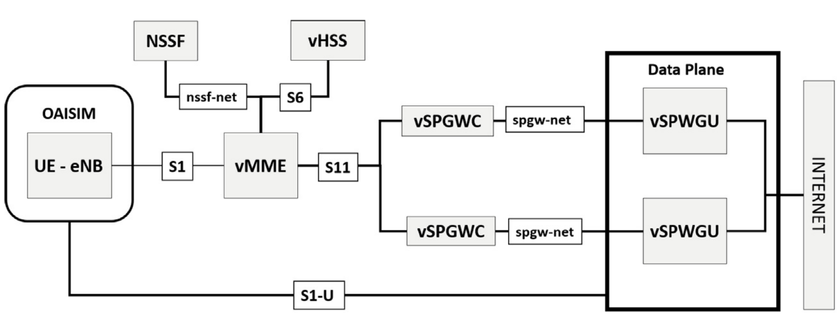

Fig. 7. The OSA's EPC software contains implementation of vMME, vHSS, vSPGW-C and vSPGW-U collectively serving as core network. The vMME connects the access network with other core network components. The vHSS is responsible for storing the subscription information of the network. The vSPGW-C manages sessions between access, core and data network. The vSPGW-U is responsible for data communication between access and data network.

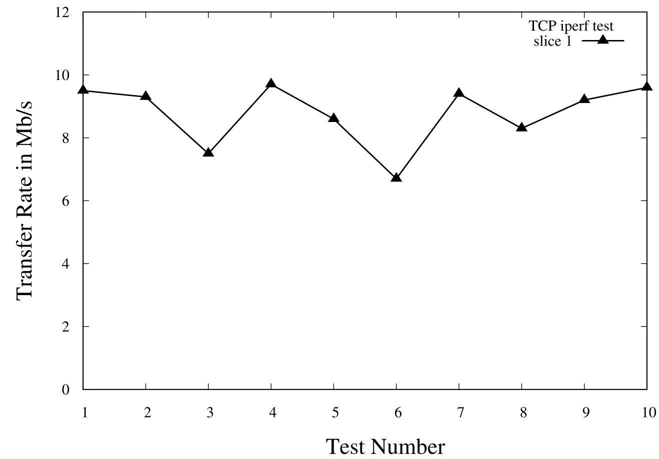

Fig. 8. Illustration of the behavior of a full E2E connection test of a UE connected to the default slice.

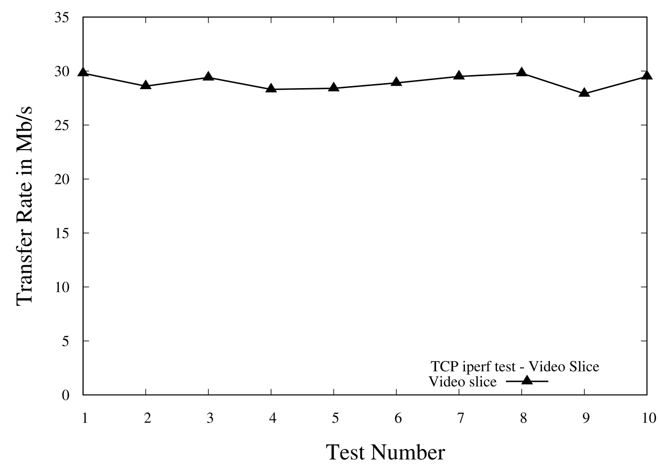

Fig. 9. VIDEO-SLICE test.

consistent results, never going above or under the 30 Mb/s range as shown in Fig. 9. Both of these tests are done consecutively.

To show the real benefit of network slicing, traffic test is generated on both slices at the same time. As iperf is not designed to run simultaneous testing, speedtest-cli is used. It works as a Linux terminal version of the speed test Web App that is widely used for testing internet bandwidth. It chooses the nearest server and conducts DL/UL Internet connection test. It is important to consider the maximum throughput that the platform can handle, in order to avoid resource conflict when using both slices for traffic. In this case, 100 Mb/s is the top limit of our test-bed. Taking this into account, if the VIDEO — slice throughput is tone down to 20 MB/s (More than enough for transmitting HD Video) both of the slices can achieve maximum performance without disrupting each other. The benefit of network slicing can be seen in the results of Fig. 10.

Hence, configured slice selection scenario is implemented in M-CORD using different NF's deployed as VNF's. Moreover, multiple logical slices are established over one physical network, where proposed NSSF selects appropriate slice based on the end user's information. Results show the allocation of independent resources to each UE specific to its needs.

It is important to consider the maximum throughput that the platform can handle, in order to avoid resource conflict when using both slices for traffic. In this case 100 Mb/s is the top limit of our test-bed. Taking this into account, if the VIDEO — slice throughput is tone down to 20 MB/s (More than enough for transmitting HD Video) both

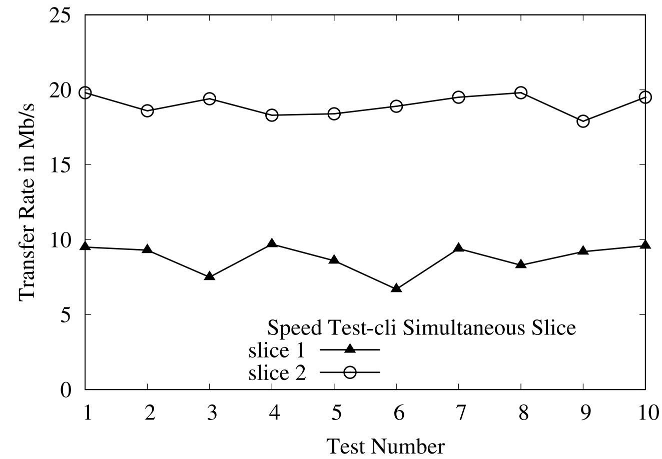

Fig. 10. Simultaneous slice test.

of the slices can achieve maximum performance without disrupting each other. The benefit of network slicing can be seen in the results of Fig. 10.

Hence, configured slice selection scenario is implemented in M-CORD using different NF's deployed as VNF's. Moreover, multiple logical slices are established over one physical network, where proposed vNSSF selects appropriate slice based on end user's information. Results show the allocation of independent resources to each UE specific to its needs.

6. Conclusions

This paper explains a network slice selection scenario using vNSSF and OAI framework under M-CORD. It introduces programmability to both control and data plane. It explains development of customized vNSSF as a VNF. With the results, it is demonstrated that the proposed network slicing mechanism successfully provides logically different resources to different users over a single physical network. In the future, we aim to extend the proposed framework by introducing intent-based networking (IBN) to the 5G architecture that will lead to automatic configuration of network slices using the intents provided as contracts. In addition, further experimental work will be carried out to achieve high data rates.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Iqbal, M.A. Iqbal, A. Ahmad, M. Khan, A. Qamar, K. Han, Comparison of spectral efficiency techniques in device-to-device communication for 5G, IEEE Access 7 (2019) 57440–57449.

[2] J. Iqbal, M. Khan, M. Afaq, A. Ali, Performance analysis of vertical handover techniques based on IEEE 802.21: Media independent handover standard, Trans. Emerg. Telecommun. Technol. (2019) e3695.

[3] B. Blanco, J.O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P.S. Khodashenas, L. Goratti, M. Paolino, et al., Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN, Comput. Stand. Interfaces 54 (2017) 216–228.

[4] P. Rost, C. Mannweiler, D.S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, et al., Network slicing to enable scalability and flexibility in 5G mobile networks, IEEE Commun. Mag. 55 (5) (2017) 72–79.

[5] S.-J. Seok, A. Muhammad, K. Kim, D. Choi, Y.-H. Han, W. Seok, W.-C. Song, multiFIA Multi-dimensional future Internet architecture, in: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2017, pp. 87–92.

[6] http://opencord.org/wp-content/uploads/2017102/MWC-M-CORD-WP-Short-Final.pdf, (Accessed 05 February 2019).

[7] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, W. Snow, Central office re-architected as a data center, IEEE Commun. Mag. 54 (10) (2016) 96–101.

[8] L. Peterson, S. Baker, M. De Leenheer, A. Bavier, S. Bhatia, M. Wawrzoniak, J. Nelson, J. Hartman, Xos: An extensible cloud operating system, in: Proceedings of the 2nd International Workshop on Software-Defined Ecosystems, ACM, 2015, pp. 23–30.

[9] https://gitlab.eurecom.fr/oai/openairinterface5G/wikis/home, (Accessed 05 March 2019).

[10] M.T. Abbas, A. Muhammad, W.-C. Song, SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach, J. Ambient Intell. Human. Comput. (2019) 1–16.

[11] M.T. Abbas, A. Muhammad, W.-C. Song, Road-aware estimation model for path duration in internet of vehicles (IoV), Wirel. Pers. Commun. (2019) 1–24.

[12] C. Campolo, R. Fontes, A. Molinaro, C.E. Rothenberg, A. Iera, Slicing on the road: Enabling the automotive vertical through 5G network softwarization, Sensors 18 (12) (2018) 4435.

[13] C. Campolo, A. Molinaro, A. Iera, R.R. Fontes, C.E. Rothenberg, Towards 5G network slicing for the V2X ecosystem, in: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), IEEE, 2018, pp. 400–405.

[14] H. Khan, P. Luoto, S. Samarakoon, M. Bennis, M. Latva-Aho, Network slicing for vehicular communication, Trans. Emerg. Telecommun. Technol. (2019) e3652.

[15] A. Muhammad, W.-C. Song, A novel framework for resource orchestration in openstack cloud platform, KSII Trans. Internet Inf. Syst. 12 (11) (2018).

[16] K. Katsalis, N. Nikaein, A. Huang, JOX: An event-driven orchestrator for 5G network slicing, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–9.