A novel QoS framework for network slicing in 5G and beyond networks based on SDN and NFV

Zhaogang Shu and Tarik Taleb

***Abstract*—Along with the development of 5G, Network Slicing (NS) plays an important role in the application of mobile networks to meet all kinds of personalized requirements. In terms of NS concept, network operators can vertically split a physical network into multiple logically separate networks to flexibly meet Quality of Service (QoS) requirements, which are mainly represented as higher bandwidth and lower latency. In this paper, we propose a novel QoS framework of NS in 5G and beyond networks based on Software Defined Network (SDN) and Network Function Virtualization (NFV) to guarantee key QoS indicators for different application scenarios, such as enhanced Mobile Broad-Band (eMBB), massive Machine-Type Communications (mMTC) and Ultra-Reliable and Low-Latency Communications (URLLC). In this QoS framework, 5G network is divided into three parts, Radio Access Network (RAN), Transport Network (TN) and Core Network (CN) to form three types of NS with different network resource allocation algorithms. The performance evaluation in the simulation environment of Mininet shows that the proposed QoS framework can steer different flows into different queues of Open Virtual Switches (OVS), schedule network resources for various NS types and provide reliable End-to-End (E2E) QoS for users according to preconfigured QoS requirements.**

***Index Terms*—QoS, Network Slicing, SDN, NFV, 5G and beyond.**

I. Introduction

ALONG with the development of the 5G network communication technology, global telecom operators started deploying 5G, which is considered as a revolutionary mobile communication system. Compared with 4G, 5G is expected to provide higher bandwidth, lower End-to-End (E2E) latency, and more flexible and reliable network access [1]. For example, it can support stable network connection for highly mobile objects and high-density distributed sensors, which are necessary for many applications of Internet of Thing (IoT). In addition to these features, the most valuable point of 5G is the possibility to bring huge business opportunities through customizing services in terms of specific requirements for different verticals, such as manufacturing, automotive and health-care industry. To achieve this goal, the concept of network slicing is adopted in 5G. The core idea beneath 5G is to divide a single physical network into multiple E2E logically-separated sub-networks, each of which is called a Network Slice (NS). Specifically, every NS has its own management domain and E2E logical topology. Operators can flexibly create, modify or destroy a NS as per different QoS requirements without disrupting other existing NS.

The key enabling technologies of NS are Software Defined Networking (SDN) and Network Function Virtualization (NFV) [2]. On one hand, SDN provides a controller-centered network management mode through the separation of control plane and forward plane, which enables network administrators to flexibly and remotely program their networks. On the other hand, NFV is a kind of abstraction mechanism to virtualize the network resources, similar with the virtualization of computing and storage resources in the cloud. In this case, network physical nodes and links can be shared by multiple separate virtual networks so that they simultaneously run on top of a common physical infrastructure. Furthermore, a virtual machine (VM) in remote cloud can be also designed as a Virtual Network Function (VNF), which works like a traditional hardware network device, such as a router, a switch, or a firewall. Therefore, integrating SDN and NFV into NS is an ideal way to meet the flexibility, reliability and scalability of various heterogeneous 5G services, which have been classified as three main groups: enhanced Mobile Broad-Band (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low-Latency Communications (URLLC) according to ITU-T [3].

Indeed, different 5G services have different QoS requirements in terms of bandwidth, latency (jitter), packet loss rate and reliability. Thus, establishing QoS-sensitive NS is one of the most critical and challenging tasks for network slicing in 5G and beyond networks. Many researchers have studied the QoS issues in the traditional network environment and proposed many solutions to improve QoS properties of networks. However, existing solutions can not be direclty applied in the NS architecture in 5G due to the increasing network heterogeneity and implementation complexity based on SDN and NFV. Some efforts have been made to address this problem [4][5], but this is still an open issue.

This paper focuses on the design and implementation of a QoS-aware network slicing framework to support 5G and beyond services and that is while leveraging SDN and NFV. The envisioned QoS framework divides 5G into three parts: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). Each part is managed by a specific SDN controller, which has a global view of the local network topology and the network status. The decision of E2E connection for NS is made in a hop-by-hop mode through collaboration amongst SDN controllers. We implement a prototype of the

Zhaogang Shu is with the Comupter and Information College and Key Laboratory of Smart Agriculture and Forestry,Fujian Agriculture and Forestry University, China. e-mail: zgshu@fafu.edu.cn

Tarik Taleb is with the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland, Information Technology and Electrical Engineering, Oulu University, Finland and the Department of Computer and Information Security, Sejong University, South Korea. e-mail: Tarik.Taleb@aalto.fi

proposed QoS framework using ONOS as a SDN controller and evaluate its performance in a Mininet-based simulation environment. The obtained results show that the proposed QoS framework can effectively schedule network resources for various NS types and provide reliable E2E connection service for users according to preconfigured QoS requirements.

The remainder of this paper is organized as follows. Section II describes the relevant research work and compares them with our proposed framework. In Section III, the proposed QoS framework is presented. Section IV introduces a prototype implementation of the proposed framework using ONOS and OVS, and evaluates its performance using Mininet. Finally, some concluding remarks and future research directions are given in Section V.

II. Related Work

In this section, we will introduce some state of the art work on QoS to improve the programmability and flexibility of networks and discuss the features of the proposed solutions. We then compare them with the proposed framework based on the identified features. To better compare these solutions, we categorize them into three groups, namely QoS solutions based on SDN, QoS frameworks to support specific network applications, and QoS solutions for network slicing in 5G.

A. QoS solutions based on SDN

Since SDN has been seen as a promising network technology for 5G, SDN-based QoS issues have equally received much attention. Generally, there is a function module of QoS in SDN controller to implement network resource monitoring and scheduling. For example, Tomovic et al. presented a controller framework with QoS provisioning for multimedia applications [6]. In this framework, four key function blocks (i.e., resource monitoring, route calculation, call admission control and resource reservation) were integrated into the controller to implement QoS management. Dutra et al. [7] proposed a solution that enabled the E2E QoS based on multi-path routing in SDN. This solution allowed operators to allocate network resources through the feature of queue in OpenFlow so that over-provisioning of bandwidth resources can be reduced or eliminated. Pan et al. proposed a programmable packet scheduling framework OpenSched [8], which was a layered architecture to glue the QoS applications, the controller and the switches together, including flexible northbound interface, controller-switch interaction and efficient southbound protocol handling, as well as QoS policy execution at the switch side. A prototype based on ONOS and OVS showed that it can facilitate flexible network resource provisioning.Oliveira et al. [9] proposed a QoS provisionning architecture to support classification of services and negotiation of QoS requirements between applications and the SDN controller, which can monitor and optimize network performance on demand and in a timely fashion.

B. QoS solutions to support specific network applications

There are some research work that concentrate on the QoS solutions for specific network applications, such as cloud data center network, smart grid network, energy network and remote medical network. Tajiki et al. studied the QoS optimization with minimum network reconfiguration overhead in the cloud data center [10]. A forwarding table compression technique was designed to implement resource reallocation, which can be deployed as an application module in the SDN controller. The experiment results showed that it efficiently decreased the network reconfiguration overhead while satisfying the QoS requirements. In the work of [11], the authors proposed a QoS model based on SDN for smart grid network. In this model, a content-aware queuing algorithm was devised so that traffic flows were categorized into different groups, which finally provided low latency connection for smart grid network. Qiu et al. [12] proposed a QoS-enabled load scheduling algorithm based on reinforcement learning for smart grid and energy network. The feature of this algorithm was to solve the problem of the cooperation among multiple controllers using Artificial Intelligence (AI) technology so that they can automatically negotiate QoS parameters. A QoS-sensitive application for medical system is introduced in [13]. The authors proposed a multi-path routing algorithm to ensure QoS requirements and improve QoS of medical information transmission in OpenStack environment using the OpenContrail controller.

C. QoS solutions for network slicing in 5G

Currently, there are also some research work focusing on QoS to support network slicing in 5G. For instance, Rafael et al. studied the Quality of Experience/Quality of Service (QoE/QoS) of 5G-enabled optical networks [4], which focused on the E2E service delivery. An architecture of NS provisioning with QoS guarantee was presented, supporting 5G service chaining in cross-domain optical networks. A policy-based monitoring and actuation framework was used to maintain the desired QoS requirements for E2E network slice. However, this framework did not provide the interaction mechanism between SDN controllers and NFV entities to make QoS decision in the context of NS when the network topology changed. A.Sgambelluri et al. [5] presented a solution to establish E2E connection with QoS constraints based on the 5G Exchange (5GEx) project through connecting VNFs deployed in remote data centers. In this solution, a stateful backward recursive path procedure was used to maintain the E2E connection services. Experiment results showed that this solution can support automatic establishment of QoS-based E2E connection across multi-operator network domains. However, this orchestration scheme was not flexible enough to support the scalability for the advertisement of resources and dynamic connection services. Vincenzi et al. [14] provided a thorough discussion of the challenges that network slicing brings in the different network parts and designed a cooperative game to study the potential cooperation aspects among the participants. Sattar et al.[15] addressed the question of optimal allocation of a slice in 5G core networks by tackling two challenges, namely function isolation and guaranteeing end-to-end delay for a slice. However, SDN and NFV technologies were not applied in these solutions.

TABLE I: Comparison between existing solutions and our proposed framework.

Solution group / Feature list	QoS solutions based on SDN [6]–[9]	QoS solutions to support specific network applications [10]–[13]	QoS solutions for NS in 5G [4], [5][14][15]	Our proposed framework
Programmability based on SDN	Y	Y	Y	Y
Dynamic multi-path routing	Y (only [7])	Y(only [13])	N	Y
Cooperation of multiple SDN controllers	N	N	N	Y
NFV in cloud	N	Y(only [10])	Y(only [4])	Y
NS in 5G and beyond networks	N	N	Y	Y
Algorithms based on AI	N	Y(only [12]	N	N

D. Comparison among the different solutions

To figure out the advantages and disadvantages of existing solutions (in comparison to our proposed framework), we compare them by checking if each solution supports different features. Some typical features include programmability based on SDN, dynamic multi-path routing, cooperation of multiple SDN controllers, NVF in cloud, NS in 5G and algorithms based on AI. TABLE. 1 compares existing solutions against our proposed framework based on these features in detail. In this table, 'Y' means the solution supports corresponding feature and 'N' means the opposite. However, not all the solutions in the same group support a specific feature. For example, in the first solution group, only the work of [7] supports the feature of 'dynamic multi-path routing'. From this comparison, we can conclude that our proposed framework covers most of the key features except 'algorithms based on AI', which indicates that it is a more comprehensive solution than other existing solutions.

III. Proposed QoS Framework for NS in 5G

In this section, we will describe our proposed QoS framework for NS in 5G and that is based on SDN and NFV. Fig. 1 depicts the envisioned QoS framework. We divide the framework into three abstract layers, namely physical network resources layer, SDN and NFV-based management layer, and QoS-sensitive slicing layer.

A. Physical network resources layer

In general, a 5G network infrastructure consists of three parts: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). End terminals (e.g. mobile phones, webcams, smart industrial devices, and vehicles) connect to 5G through the wireless base station in RAN. Actually, in 5G, RAN nodes may collocate with nodes offering computing and storage resources, forming the so-called Multi-Access Edge Cloud (MEC). In this case, many network functions can be implemented as Virtualized Network Functions (VNF) in MEC, which also enables the softwarization of RAN. TN locates between RAN and CN. Similar to traditional Metropolitan Area Network (MAN) and Wide Area Network (WAN), TN geographically covers several kilometers even longer distances to connect different RANs and CNs. To enhance the functions and the management of TN, a cloud data center may be also built to support SDN and NFV, which are responsible for WAN optical network management, mobile management and user data analysis. CN represents the core network that manages all VNFs and Physical Network Functions (PNFs), forming a single mobile operator network, as well as connectivity to/from end users through these VNF/PNFs to access the Internet and other mobile services.

B. SDN and NFV based management layer

SDN and NFV based management system is designed in this layer to manage the physical resources, including network devices, computing resources and storage resources. Each part of the physical networks (RAN, TN and CN) has its own management system, which can communicate to each other through their corresponding SDN controllers. We call the physical resources of RAN Level1 Physical Resource Pool(L1PRP for short). Similarly, the physical resources of TN and CN are called L2PRP and L3PRP, respectively. On top of these resource pools, there exists a virtualization layer which is used to divide the physical resources into logically independent VNFs. According to European Telecommunications Standards Institute (ETSI), VNF Management and Orchestration contains two components: Virtualization Infrastructure Manager (VIM) and VFN Manager (VNFM). VIM and VNFM interact with each other to manage the life cycle of VNFs, such as creating, migrating, modifying, and destroying VNFs. A SDN controller is also deployed to manage the connection among these VNFs through interacting with VNFM. The SDN controller in RAN is called Level1 SDN Controller (L1SC). Likewise, the SDN controllers in TN and CN are called L2SC and L3SC, respectively. L1SC is responsible for reporting local network information to L2SC, while L2SC is responsible for reporting

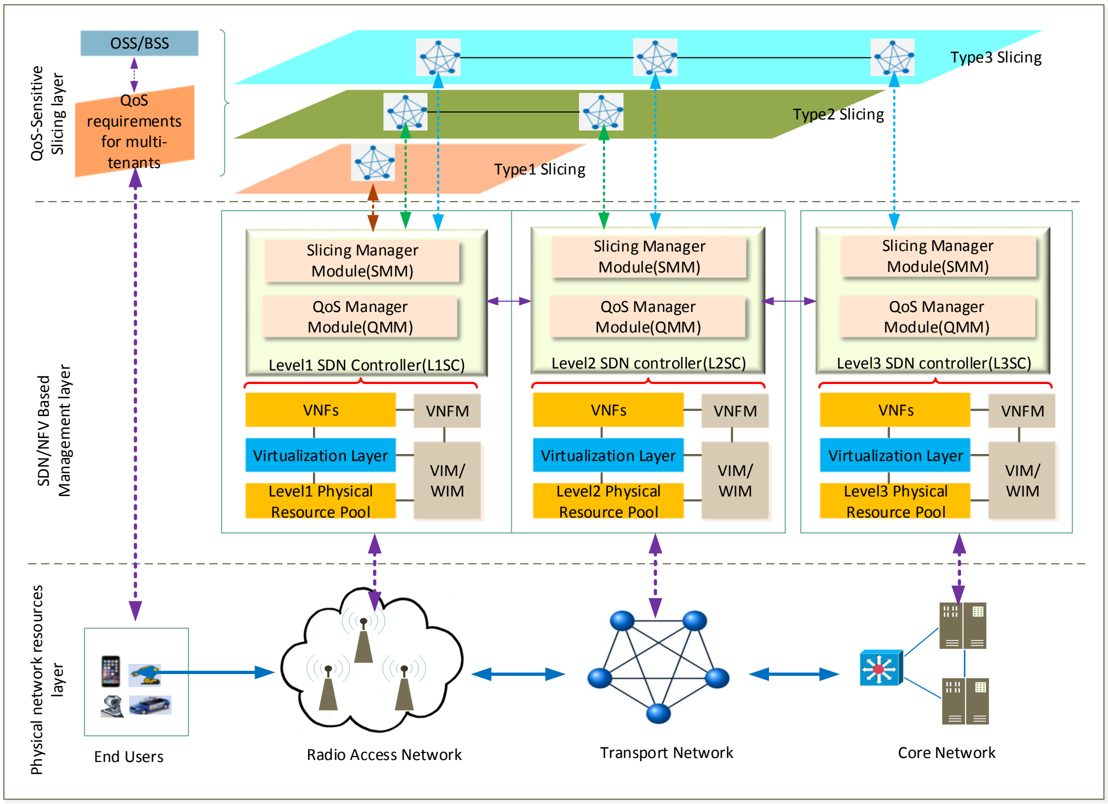
OSS/BSS
QoS requirements for multi-tenants
QoS-Sensitive Slicing layer
Type3 Slicing
Type2 Slicing
Type1 Slicing
SDN/NFV Based Management layer
Slicing Manager Module(SMM)
QoS Manager Module(QMM)
Level1 SDN Controller(L1SC)
Level2 SDN controller(L2SC)
Level3 SDN controller(L3SC)
VNFs
VNFM
Virtualization Layer
VIM/ WIM
Level1 Physical Resource Pool
Level2 Physical Resource Pool
Level3 Physical Resource Pool
Physical network resources layer
End Users
Radio Access Network
Transport Network
Core Network

Fig. 1: Proposed QoS framework for NS in 5G based on SDN and NFV.

local network information to L3SC. Finally all L3SCs must synchronize all network information to keep the consistency of the whole network. Fig. 2 illustrates the mechanism of this kind of distributed hierarchical SDN controllers architecture.

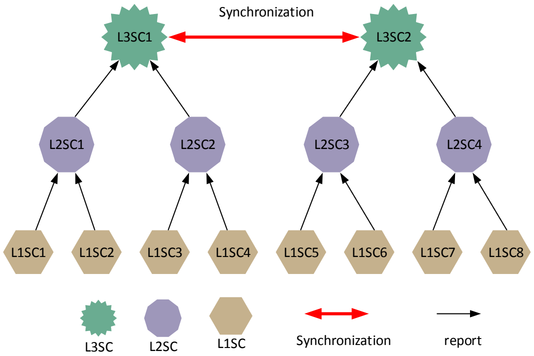

Fig. 2: Distributed Hierarchical SDN Controllers Architecture.

In this situation, each L1SC in RAN has a limited topology view of one RAN domain and just communicates with corresponding L2SC in TN without caring about other L1SCs. In the same way, each L2SC in TN has also a limited topology view of one TN domain and just needs to report its information to corresponding L3SC in CN without caring about other L2SCs. For example, if two end users access the same RAN, the communication between them can easily be handled by just the L1SC in this RAN. However, when the end users are located in different RANs, L2SC must be involved to allocate related network resources, cooperating with two L1SCs. Furthermore, when the end users are distributed across different TNs, L3SC may also take part in the process of network resources allocation. Hence the quantity of SDN controllers can be scaled up or down flexibly according to the current network size, by which we can greatly improve the scalability of network management.

It should be pointed out that all SDN controllers (L1-L3) here should be modified to adapt the features of NS in 5G. Effectively, in addition to the basic SDN controller features (e.g., topology management and OpenFlow-based routing management), each SDN controller also contains two key modules, Slicing Manager Module (SMM) and QoS Manager Module (QMM). SMM manages the life cycle of all network slices created in its respective network domain, while QMM is in charge of the allocation and scheduling of network resources therein.

C. *QoS-sensitive slicing layer*

In this layer, the Operation Support System (OSS) and Business Support System (BSS) interact with SMM/QMM modules of SDN controllers to realize E2E NS along with the QoS requirements of multi-tenants. Actually, OSS/BSS can be considered as an application module of SDN controllers, since most of industrial SDN controllers (e.g. ONOS and OpenDaylight) provide RESTfull north-bound application programming interface. Diverse network slices can be customized by this interaction model to realize desired QoS service. According to the scope of network resources needed by NS, we define three types of NS. Type1 NS works in a single RAN, implicating that packet forwarding paths for all end users are constrained within the RAN. Type2 NS works with the packet forwarding path like RAN-TN-RAN, while the packet forwarding path of Type3 NS is like RAN-TN-CN-TN-RAN or its sub-path. When tenants rent a NS, they should consider both QoS requirements and the cost to decide which type NS is the best choice for them.

IV. Experimental evaluation

In this section, we fist introduce the experimental environment and then validate the feasibility of the proposed framework through evaluating the performance of network resources allocation algorithms of Type1 and Type2 NS. Based on the analysis of experiment results, we give some general conclusions of network planning for NS in 5G.

A. Experiment environment Setup

ONOS is a highly-modular distributed SDN controller, which can be deployed in a large-scale network to form a cluster of controllers. Each controller manages part of the network nodes, communicating with other controllers to keep the state consistency of the whole network timely. In addition, ONOS provides many flexible north-bound RESTful interfaces to allow operators to develop new application modules, which can be easily integrated into ONOS. In order to evaluate the feasibility of the proposed QoS framework, we develop a prototype system as an application module of ONOS, which is composed of two network resources allocation algorithms and common functions. As shown in Fig. 3, we consider two scenarios to simulate the process of creating Type1 NS and Type2 NS with different QoS requirements, respectively. Algorithm1 is designed to compute forwarding path from source node to destination node and allocate network resources to meet QoS requirements for Type1 NS, while Algorithm2 has the same function for Type2 NS. Generally, the packet flow is recognized by the pair of source IP address and destination IP address, so we can take it as a NS in this context. Common functions focus on two tasks: 1) collecting the bandwidth and latency data between any two OVS nodes that connect directly in Mininet; 2) steering a specific flow into a specific queue of OVS port through modifying the flow table in OVS. Based on the application module of ONOS, we conduct the experiments on two computers with Intel multi-core i5-4300 CPU and 8G RAM. The operating systems of two computers are both Ubuntu 18.04 LTS and they are connected directly through network interface with the speed of 1Gbps. ONOS (version 1.15.0) runs on one computer as SDN controller and Mininet (version 2.3.0d) /OVS (version 2.9.2) runs on the other computer as network topology simulation environment.

B. Performance evaluation

When we create a new NS in the running network, the algorithms will check the available resources (links and bandwidth) in the network and try to find one or multiple forwarding paths from source node to destination node to satisfy the QoS requirements of this NS. In our algorithms, we define four parameters as the inputs of algorithms, including IP address of source node, IP address of destination node, maximum bandwidth and minimum latency. At the beginning, there are enough available links and bandwidth in the network, so the algorithms can find a forwarding path for a new NS very quickly. Along with a decrease in the available links and bandwidth in the network, algorithms will take longer time to find the paths, and may even fail to find one. In addition, the size and type of network topology also greatly impact the execution time of the algorithms to find the path. Therefore, by investigating the processing time of creating new NS, we can observe the performance of the proposed QoS framework with different type NS and different network status, which may provide some useful insights. Type1 NS belongs to a single network domain, while both Type2 and Type3 NS belong to multiple network domains. In our design, the network resources allocation algorithm of Type3 NS is similar with that of Type2 NS. We can iterate the algorithm of Type2 NS to allocate resources for Type3 NS. Therefore, here we just evaluate the Type1 and Type2 NS algorithms as typical examples.

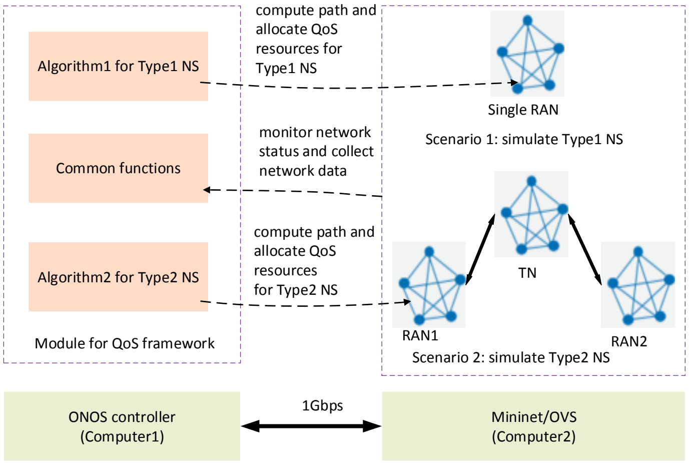
Algorithm1 for Type1 NS
compute path and allocate QoS resources for Type1 NS
Single RAN
Scenario 1: simulate Type1 NS
Common functions
monitor network status and collect network data
Algorithm2 for Type2 NS
compute path and allocate QoS resources for Type2 NS
TN
RAN1
RAN2
Scenario 2: simulate Type2 NS
Module for QoS framework
ONOS controller (Computer1)
1Gbps
Mininet/OVS (Computer2)

Fig. 3: Experiment Environment and Scenarios.

We first consider a scenario of creating 100 Type1 NS continuously in two networks that contain 20 and 100 OVS nodes, respectively. These networks are designed as full-mesh networks whereby any two nodes connect directly. The bandwidth and latency of every link of network are generated randomly with constraints of bandwidth $\in [50Mbps, 100Mbps]$ and latency $\in [1ms, 10ms]$. For each Type1 NS request, let $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ denote the inputs of our algorithm to create NS, in which EU_s,EU_d,B_{min} and L_{max} stand for IP address of source node, IP address of destination node, maximum bandwidth and minimum latency required by this NS, respectively. R_{NS} is generated randomly with constraints of $B_{min} \in [1Mbps, 5Mbps]$ and $L_{max} \in$

$[50ms, 100ms]$, which also indicates that the positions of source node and destination node are specified randomly. Fig. 4 presents the processing time for Type1 NS request in this scenario.

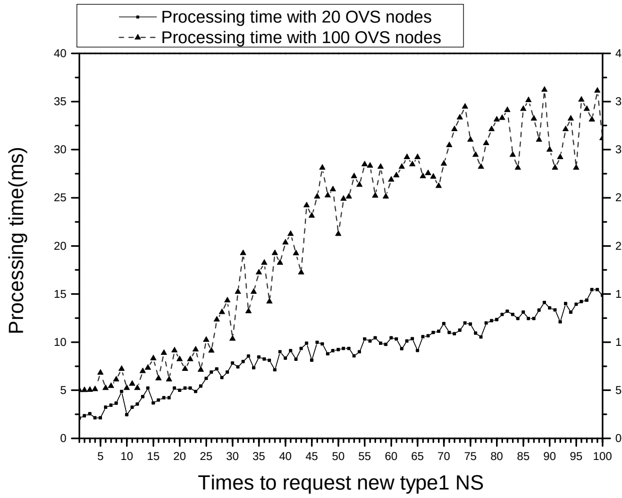

Fig. 4: The processing time for new Type1 NS with constraints of $B_{min}\in [1Mbps, 5Mbps]$ and $L_{max} \in [50ms, 100ms]$.

In Fig.4, the left Y-axis is the processing time in millisecond and the bottom X-axis is the number of requests to create Type1 NS, varied from 1 to 100. The straight line represents the processing time in the network with 20 OVS nodes while the dot line shows the processing time in the network with 100 OVS nodes. To describe conveniently, we call the network with 20 OVS nodes as network A and the network with 100 OVS nodes as network B. We can see that as the number of requests increases, the processing time of network A increases from $2.5ms$ to $15ms$ in an approximately linear rate. It shows that the available network resources in the same network are becoming fewer, SDN controllers will take longer time to create a new NS, because the forwarding path of the new NS includes more OVS nodes, even sometimes needs multiple sub-paths to satisfy the requirements. We also notice that the processing time of network B increases faster than that of network A. As expected, the reason is that the target path in network B becomes more complicated than that of network A along with the decrease of network resources in it. Furthermore, there is a little fluctuation for both straight and dot lines, which means, sometimes a new request probably requires less time than the previous request. For example, the processing time of the $74th$ request in network B is about $35ms$ while that of the $77th$ request is about $28ms$. Then we set the QoS constrains as $B_{min} \in [5Mbps, 10Mbps]$ and $L_{max} \in [10ms, 20ms]$ and run 100 NS requests in the same network. As shown in Fig. 5, it is obvious that the processing time is longer than that experienced in the previous scenario. The processing time of the previous scenario is less than $40ms$ while the maximum processing time in this scenario is close to $100ms$. From these results, we can notice that B_{min} and L_{max} should be the significant factors to affect the processing time of algorithms to find a forwarding path.

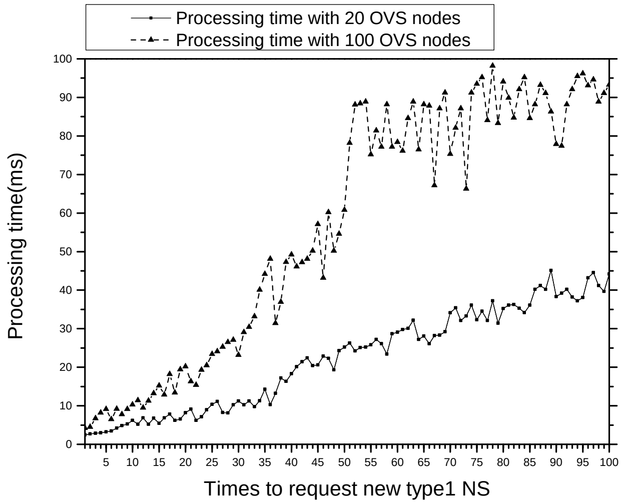

Fig. 5: The processing time for new Type1 NS with constraints of $B_{min} \in [5Mbps, 10Mbps]$ and $L_{max} \in [10ms, 20ms]$.

Like the evaluation method of Type1 NS, we conduct experiments of requesting new Type2 NS in a more complicated network topology that contains three full-mesh networks: RAN1, TN and RAN2. RAN1 and TN are connected by only two links, which belong to two edge nodes in the network. Similarly, TN and RAN2 are also connected by two links directly. We set the bandwidth and latency of links between two full-mesh networks to $200Mbps$ and $1ms$, respectively. The bandwidth and latency of every link in three full-mesh networks are generated randomly. For each Type2 NS request, the inputs of algorithms $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ are also generated randomly with the constrains $B_{min} \in [1Mbps, 5Mbps]$ and $L_{max} \in [50ms, 100ms]$, which also means the positions of source node and destination node are specified randomly in RAN1 and RAN2. Fig. 6 presents the relationship between processing time and number of requests for Type2 NS in this scenario.

As described before, we call the network with 20 OVS nodes as Network A, which means each network domain (RAN1,TN and RAN2) contains 20 OVS nodes in this scenario. In the same fashion, each network domain of Network B contains 100 OVS nodes. We can see that the remarkable feature of processing time for Type2 NS is that there is a sudden down in the middle, where the number of requests is between 40 and 50. For Network A, the processing time decreases from about $60ms$ to $18ms$ at the 50th request. For Network B, the processing time decreases from about 140ms to $25ms$ at the $40th$ request. The main reason is that the bandwidth of one link between two network domains is used up. In this situation, the underlying algorithm will choose another link belonging to another edge node and find an E2E path very quickly, which causes the decrease in processing time. For instance, the bandwidth of one link between two network domains is $200Mbps$, and the QoS requirements of

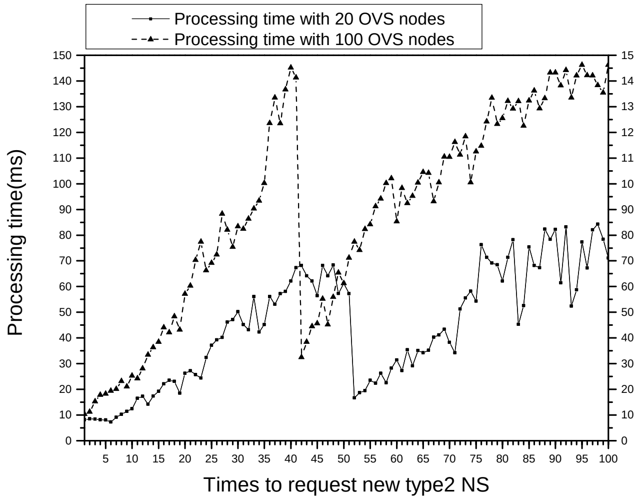

Fig. 6: The processing time for new Type2 NS with constraints of $B_{min} \in [1Mbps, 5Mbps]$ and $L_{max} \in [50ms, 100ms]$.

NS is $B_{min} \in [1Mbps, 5Mbps]$, therefore it is reasonable to use up all bandwidth of one link when the request times is about 40 to 50. Fig. 7 shows the processing time when the QoS requirements are set to $B_{min} \in [5Mbps, 10Mbps]$ and $L_{max} \in [10ms, 20ms]$ in the same network topology.

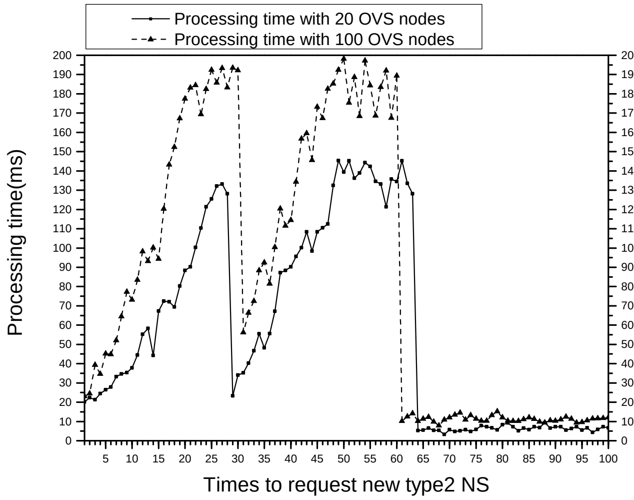

Fig. 7: The processing time for new Type2 NS with constraints of $B_{min} \in [5Mbps, 10Mbps]$ and $L_{max} \in [10ms, 20ms]$.

We observe that the processing time of Fig. 6 is less than 150ms while the maximum processing time in Fig. 7 is close to 200ms. For the same reason, there is also a sudden down for the processing time of Network A and Network B when the number of requests is about 30. We also notice that when the number of requests of new Type2 NS is getting to about 60 or 65, the processing time decreases to about 10ms sharply. After checking the iterations of the algorithm, we found that the reason was the failure of requesting new NS due to not enough bandwidth of the links between the two network domains. The bandwidth of one link between two network domains is $200Mbps$, while the constraint of NS requirements is $B_{min} \in [5Mbps, 10Mbps]$, so it is reasonable to use up all the bandwidth after the 30th NS requests.

Generally speaking, the processing time of an algorithm is tightly correlated with the number of its iterations to find available paths to meet the QoS requirements, and many factors can impact the number of iterations of algorithms, such as the network topology, the positions of end users, and the QoS requirements. Therefore, we can get some general conclusions from the processing time when we create a new NS. First, for Type1 NS in a single network domain, the available network resources are the main factor to create a NS successfully if the end users are located randomly. Second, for Type2 or Type3 NS in multiple network domains, the location and number of edge nodes in each network domain are also significant factors to create a NS. In our experiment, the sudden decrease in processing time of algorithms is caused by the switching from one link to another link between two network domains, and the edge nodes containing these two links are located very nearly to each other. Therefore, we should try to scatter the edge nodes of network domains to avoid this situation when designing multiple domain networks.

V. Conclusion

This paper proposes a QoS framework for network slicing in 5G and beyond networks based on SDN and NFV. Through dividing networks into three parts, namely RAN, TN and CN, we describe the function modules of the QoS framework in details from the perspectives of three layers: physical network resources layer, SDN and NFV based management layer and QoS-sensitive slicing layer, where we classify NS into three types. Based on this, we design different algorithms to schedule network resources according to the bandwidth and latency requirements for different NS. The results show that the proposed framework can create NS for users flexibly and provide useful guidance for the development of the QoS framework for NS in 5G. Combining our algorithms with AI techniques to optimize the network resources allocation for NS in 5G defines some of our future work in this area.

Acknowledgment

This research work has partially received funding from CERNET innovation project of China (NO.NGII20190102). It was also partially supported by the European Union's Horizon 2020 Research and Innovation Program through the MonB5G Project under Grant No. 871780, by the Academy of Finland 6Genesis project under Grant No. 318927, and by the Academy of Finland CSN project under Grant No. 311654.

Biographies

Zhaogang Shu (zgshu@fafu.edu.cn) received his Ph.D. degree from South China University of Technology, Guangzhou, China in 2008. He is currently an associate professor of Fujian Agriculture and Forestry University, Fuzhou, China. He has presided three advanced research projects, authored more than 15 journal papers and held four patents. His research interests include software-defined wireless networking, network security and mobile edge computing.

Tarik Taleb (Tarik.Taleb@aalto.fi) received the B.E. degree (with distinction) in information engineering in 2001, and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2003 and 2005, respectively. He is currently a professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. He is the founder and the director of the MOSA!C Lab (http://www.mosaic-lab.org/).

References

[1] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[2] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[3] P. Popovski, K. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view. 2018."

[4] R. Montero, A. Pagès, F. Agraz, and S. Spadaro, "Supporting qoe/qos-aware end-to-end network slicing in future 5g-enabled optical networks," in *Metro and Data Center Optical Networks and Short-Reach Links II*, vol. 10946. International Society for Optics and Photonics, 2019, p. 109460F.

[5] A. Sgambelluri, O. Dugeon, A. Muhammad, J. Martín-Pérez, F. Ubaldi, K. Sevilla, O. De Dios, T. Pepe, C. Bernardos, P. Monti *et al.*, "Orchestrating qos-based connectivity services in a multi-operator sandbox," *Journal of Optical Communications and Networking*, vol. 11, no. 2, pp. A196–A208, 2019.

[6] S. Tomovic, N. Prasad, and I. Radusinovic, "Sdn control framework for qos provisioning," in *2014 22nd Telecommunications Forum Telfor (TELFOR)*. IEEE, 2014, pp. 111–114.

[7] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end qos based on multi-paths routing using sdn technology," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[8] T. Pan, T. Huang, J. Mao, C. Li, and Y. Liu, "Opensched: Programmable packet queuing and scheduling for centralized qos control," in *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, 2017, pp. 95–96.

[9] A. T. Oliveira, B. J. C. Martins, M. F. Moreno, A. B. Vieira, A. T. A. Gomes, and A. Ziviani, "Sdn-based architecture for providing qos to high performance distributed applications," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 00 602–00 607.

[10] M. M. Tajiki, B. Akbari, and N. Mokari, "Optimal qos-aware network reconfiguration in software defined cloud data centers," *Computer Networks*, vol. 120, pp. 71–86, 2017.

[11] M. Rezaee and M. H. Y. Moghaddam, "Sdn-based quality of service networking for wide area measurement system," *IEEE Transactions on Industrial Informatics*, 2019.

[12] C. Qiu, S. Cui, H. Yao, F. Xu, F. R. Yu, and C. Zhao, "A novel qos-enabled load scheduling algorithm based on reinforcement learning in software-defined energy internet," *Future Generation Computer Systems*, vol. 92, pp. 43–51, 2019.

[13] K. Venkatesh, L. Srinivas, M. M. Krishnan, and A. Shanthini, "Qos improvisation of delay sensitive communication using sdn based multipath routing for medical applications," *Future Generation Computer Systems*, vol. 93, pp. 256–265, 2019.

[14] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-tenant slicing for spectrum management on the road to 5g," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 118–125, 2017.

[15] D. Sattar and A. Matrawy, "Optimal slice allocation in 5g core networks," *IEEE Networking Letters*, vol. 1, no. 2, pp. 48–51, 2019.