INVITED PAPER *Special Section on Wired-and-Wireless Network System Technologies in Beyond 5G Era*

Resource Allocation Modeling for Fine-Granular Network Slicing in Beyond 5G Systems

Zhaogang SHU[†a)], **Tarik TALEB**[††b)], ***and*** **Jaeseung SONG**[†††c)], ***Nonmembers***

SUMMARY Through the concept of network slicing, a single physical network infrastructure can be split into multiple logically-independent Network Slices (NS), each of which is customized for the needs of its respective individual user or industrial vertical. In the beyond 5G (B5G) system, this customization can be done for many targeted services, including, but not limited to, 5G use cases and beyond 5G. The network slices should be optimized and customized to stitch a suitable environment for targeted industrial services and verticals. This paper proposes a novel Quality of Service (QoS) framework that optimizes and customizes the network slices to ensure the service level agreement (SLA) in terms of end-to-end reliability, delay, and bandwidth communication. The proposed framework makes use of network softwarization technologies, including software-defined networking (SDN) and network function virtualization (NFV), to preserve the SLA and ensure elasticity in managing the NS. This paper also mathematically models the end-to-end network by considering three parts: radio access network (RAN), transport network (TN), and core network (CN). The network is modeled in an abstract manner based on these three parts. Finally, we develop a prototype system to implement these algorithms using the open network operating system (ONOS) as a SDN controller. Simulations are conducted using the Mininet simulator. The results show that our QoS framework and the proposed resource allocation algorithms can effectively schedule network resources for various NS types and provide reliable E2E QoS services to end-users.
key words: *QoS, network softwarization, network slicing, reliable communications, SDN, NFV, 5G, beyond 5G*

1. Introduction

The ongoing advances in the 5G system have led to thriving mobile communications by targeting broad services and applications. In contrast to previous generations, which were consumer-based technologies, 5G and beyond systems, besides regular customers, target vertical industries, such as industry 4.0, virtual reality (VR), and the Internet of Things (IoT). The 5G system mainly targets three use cases [1]: *i*) Enhanced mobile broadband (eMBB) that targets regular customers by enabling high-data-rate access to mobile services and multimedia contents; *ii*) Ultra-reliable low-latency communications (URLLC) that target mission-critical connectivity verticals including, but not limited to, autonomous driving and industry 4.0, which have rigorous requirements on reliability, latency, and availability; and *iii*) Massive machine-type communications (mMTC) carrying highly dense and immensely connected devices deployed in IoT scenarios. Currently, the 5G system is being deployed in many countries, and it is time for both academia and industry to shift their attention to its successor, namely the sixth generation (6G) system. The latter should satisfy industrial verticals targeted by information and communications technology (ICT) in 2030.

Tremendous efforts have been carried out towards the softwarization of mobile networks in 5G. This has led to the popular concept of network slicing. Some level of success has been achieved in adopting IT and software engineering principles in the design of 5G network slices. The outcome is the 5G Service Based Architecture, consisting of network functions that are virtualized and instantiable on cloud/virtualization platforms. Despite the achieved progress, the current design of the 5G network architecture is yet far from being truly cloud-native. The network slices considered in the context of 5G are still static, following three types of blueprints, each defined for a specific 5G use case (i.e., eMBB, URLLC, or mIoT/mMTC). They do not reflect, nor do they meet the frequently changing requirements of provisioned services and the dynamicity of the respective end users. Furthermore, network slices, in the context of 5G, are limited to the core network; i.e., the radio access network is simply shared among multiple network slices. An important expectation at beyond 5G (B5G) is to achieve the vision of dynamic and fine-granular network slicing. To attain true customization of network slices with fine granularity, it becomes vital to perform resource allocation according to the dynamics of the underlying infrastructure, the dynamics of the provisioned services, and the spatio-temporal behavior of the end-users. This underpins the focus of this paper.

While we are in the early stage to determine all the targeted B5G/6G use cases, some initiatives have started defining the next use cases. According to [2], the ICT 2030 is expected to support use cases related to virtual and augmented reality communications, holographic-type communications, digital twin, ubiquitous intelligence, tactile Internet, and multi-sense experiences. These use cases can be categorized into four main classes: eMBB Plus (eMBBPlus), Big Communications (BigCom), Holographic and Tactile Communications (HTC), and Secure URLLC (SURLLC) [3]. The eM-

Manuscript received May 17, 2021.
Manuscript revised July 27, 2021.
Manuscript publicized October 19, 2021.
[†]The author is with the Computer and Information College, Fujian Agriculture and Forestry University, China.
[††]The author is with the Center of Wireless Communications, The University of Oulu, Finland.
[†††]The author is with the Department of Computer Security and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea.
a) E-mail: zgshu@fafu.edu.cn
b) E-mail: Tarik.Taleb@oulu.fi
c) E-mail: jssong@sejong.ac.kr
DOI: 10.1587/transcom.2021WWI0002

BBPlus aims to replace eMBB by providing a high quality of experience (QoE) and considering other services, such as accurate indoor positioning. BigCom aims to provide excellent coverage by allowing customers to communicate everywhere with high bandwidth. Meanwhile, HTC mainly targets 3D holographic communications and the tactile Internet to allow real-time transfer of senses. Last but not least, SURLLC, besides the reliability and latency, targets security. On the other hand, six different use cases have been considered in [4]. Besides the three use cases targeted by the 5G system, three new use-cases have been considered: *i*) ubiquitous MBB (uMBB), similar to BigCom, targets the ubiquitous global connectivity of eMBB; *ii*) Ultra-reliable low-latency broadband communication (ULBC) targets verticals that require high throughput in addition to reliability and reduced latency, such as collaborative virtual reality and holographic communications; and *iii*) massive ultra-reliable low-latency communication (mULC) that simultaneously considers the features of mMTC and URLLC for the benefits of industrial verticals with dense sensors and actuators.

The aforementioned use cases have challenging key performance indicators (KPIs) in terms of network bandwidth, end-to-end (E2E) latency, and network reliability and availability. These challenges call for a beyond 5G architecture with high customization and elasticity to adapt to various scenarios. Network softwarization technologies, including network function virtualization (NFV), software-defined networking (SDN), and network slicing (NS) [5], will play a crucial role in the planned architecture due to their advantages for overcoming the everyday challenges. Regarding the latter, fine granular network slicing is the cornerstone. It has been generally agreed that applications' diverse requirements can no longer be satisfied by the "one-size-fits-all" model of conventional mobile networks. NS's fundamental idea is to partition the common physical network infrastructure into multiple logically independent or virtual networks for different verticals, such as industry 4.0, autonomous driving, and smart city applications. From this perspective, NS is similar in spirit to the concept of Infrastructure as a Service (IaaS) in cloud computing. Effectively, every virtualized network function (VNF) is created, modified, and destroyed as per the NS owner or users' changing needs, and independently without disrupting other ongoing network slices [6].

While network slices are created on cloud systems that do not necessarily guarantee high reliability, it is vital for the success and wide adoption of services running on network slices to be reliable. In this vein, it is mandatory to ensure NS reliability by guaranteeing the desired QoS. Network softwarization enablers, including SDN, NFV, and NS, will play an essential role in overcoming the challenges associated with reliability [7]. By separating the network control and data planes, SDN provides a softwarization solution to flexibly change networking rules according to different user requirements and circumstances. Operators program SDN controllers to accordingly customize the functionalities of packet forwarding elements (e.g., switches and open virtual switches). Complementarily to SDN, NFV provides an abstract mechanism to virtualize the network functions, whereby traditional network devices or function blocks are realized by software and deployed on demand in places of interest (i.e., in the local server or remote cloud). Accordingly, SDN and NFV help in creating virtual networks along with VNFs on top of common physical infrastructure and managing them in a flexible and economical way [8]. In this regard, it is essential to efficiently integrate SDN and NFV to develop a reliable QoS-aware NS management framework. This defines the core objective of this paper.

Recently, quite a few research studies have addressed this topic. Some of them presented conceptual frameworks and discussed their applicability to 5G and beyond services [6], [7], [9], [10]. However, they did not discuss the details of the implementation. Some studies focused on the design of network slicing mechanisms for only a part of the 5G network, such as Radio Access Network or Core Network [11], [12]. These works developed their respective NS frameworks for parts of the 5G network and did not consider the whole network topology of 5G to meet the end-to-end (E2E) user requirements. Many other works concentrated on domain-specific applications, such as medical applications [13], multimedia use cases [14], and the Internet of Things [15]. However, these studies created virtual networks for only specific application scenarios without considering their particular QoS parameters, such as bandwidth, latency (jitter), packet loss rate, and reliability [16]. Admittedly, establishing a QoS-sensitive connection is one of the most critical and challenging tasks for NS in beyond 5G [17]. Although many studies have been conducted on QoS provisioning in traditional networks, it is not easy to apply these solutions directly in virtual network slices due to the heterogeneous nature of the underlying physical infrastructure and the implementation complexity associated with network softwarization.

To design a comprehensive solution for reliable network slicing in beyond 5G, we propose a QoS framework for fine granular NS that makes use of both SDN and NFV. An abridged version of this paper has been published in [18]. In this work, we described the fundamental concept of the envisioned framework and conducted basic experiments to validate the framework. In comparison to the work presented in [18], this study makes the following extensions. First, it derives network requirements of users and accordingly devises a beyond 5G network resource model. Second, three kinds of resource allocation algorithms for NS have been designed and developed. Third, more experiments are conducted to validate the proposed algorithms. The experiment results show the proposed algorithms' efficiency for achieving their desired objectives in terms of E2E reliability, QoS, and delay.

The remainder of this paper is organized as follows. In Sect. 2, we discuss previous research works related to network resource modeling and allocation algorithms for NS. The envisioned QoS framework is then presented in Sect. 3. The formal model of network resources in beyond 5G and the envisioned resource allocation algorithms are described in details in the same section. Section 4 presents the prototype implementation based on the ONOS controller. It also introduces and discusses the performance evaluation conducted in the Mininet environment. Finally, some concluding remarks and future research directions are presented in Sect. 5.

2. Related Work

This section will discuss different research work on networking slicing in 5G and beyond, mainly focusing on relevant network resource allocation algorithms. We organize the solutions into groups, highlighting the advantages and pitfalls of each. We also list the different features suggested by these solutions and compare them to the solution proposed herein.

2.1 QoS Frameworks Based on SDN

Research on SDN-based QoS provisioning has received considerable attention, as SDN is considered as a promising network softwarization technology. Generally, QoS provisioning is one of the crucial functionalities of the SDN technology. Thus, related works have mainly focused on developing a SDN controller module, considering traffic engineering, multi-path routing algorithms, and load balancing issues [19]–[23]. For instance, Tomovic et al. suggested a SDN controller framework with QoS provisioning for multimedia applications [19]. In this framework, four key function blocks – resource monitoring, route calculation, call admission control and resource reservation – were integrated into controllers for enabling the QoS functionality. Dutra et al. proposed a solution that enables E2E QoS based on multi-path in SDN [21]. An extension of this work was published in [20]. These solutions allow the operators to allocate network resources through the queue features in OpenFlow so that over-provisioning of bandwidth resources is reduced or eliminated. The primary purpose of these works is to obtain fine-grained and dynamic control of network resources using SDN to prevent the limitations of legacy networks. In fact, the architecture of traditional networks is not suitable to adapt to the changing requirements of users. However, most of these solutions are designed in a centralized controller in a single mode and for small scale networks. It is difficult to apply them to large-scale Beyond 5G (B5G) networks, which are known for their heterogeneity and complexity. Nevertheless, these works provide valuable guidelines for designing the QoS provisioning framework in B5G, such as network resource allocation algorithms and mechanisms for interactions with other network function modules within the SDN controllers.

2.2 QoS Frameworks to Support Specific Applications

Each network application has its own QoS requirements. Therefore, many researchers have concentrated on QoS-sensitive network solutions for specific applications, such as cloud data center networks [24], smart grid networks [25], energy networks [26], and Internet of Things [27]. Besides, Chekired et al. designed a framework to enhance the QoS of an autonomous driving application [11]. This framework contained a network and an NS system model to improve transmission efficiency and satisfy the low latency constraint. Bektas et al. provided an E2E solution by allocating RAN network resources, creating NS for mission-critical applications via a novel scheduler [12]. A common feature of all these works is their focus on a single application domain, hence rendering them not applicable to other applications with different requirements.

2.3 QoS Support for Reliable NS

There have been studies on QoS provisioning for network slices [28]–[30]. These studies make use of the programmability of both control and data planes. Baldini et al. [31] described how the control plane of SliceNet is involved in meeting the E2E needs of different 5G verticals. Ricart-Sanchz et al. [32] proposed a QoS-aware NS framework based on hardware acceleration to support data plane programmability [32]. Another important issue is resource deployment, and management policy for E2E NS [33], [34]. Solutions proposed in [10], [35] leveraged edge/fog computing for optimizing the E2E NS. For instance, Lu Ma et al. [36] proposed a framework to manage 5G services by leveraging NFV, SDN and edge computing. The framework was designed to provide distributed and on-demand deployment of network functions and create service-guaranteed network slices based on optimal workload allocation. As a promising technology, artificial intelligence (AI) was also explored to manage network resources [37], [38].

2.4 Solutions Comparison

To recognize the advantages and disadvantages of existing solutions and compare them against our envisioned framework, we use some typical features. These features include programmability, i.e., thanks to the flexibility offered by SDN, dynamic multi-path routing, cooperation of multiple SDN controllers, NFV in cloud, edge computing, NS and AI-based algorithm. Table 1 depicts a comparison between the proposed solutions and the related works with respect to the aforementioned features. In the table, ‘Y’ means that a solution supports or offers that feature, while ‘N’ represents the opposite. It shall be noted that not all solutions in the same group support a specific feature. For example, in the solution group of “QoS Service or framework based on SDN”, only the solution presented in [21] has the feature of dynamic multi-path routing. From this comparison, we state that our proposed framework covers most of the typical features except ‘AI-based algorithm’. This demonstrates the added value of this work compared to the existing ones.

Table 1 Comparison between existing solutions and our proposed framework.

feature list \ solution group	QoS services or framework based on SDN [19], [21]–[23]	QoS framework to support specific applications [11]–[13], [15], [24]–[27]	QoS services for the NS in 5G [10], [28]–[35], [37], [38]	our proposed framework
programmability based on SDN	Y	Y	Y	Y
dynamic multi-path routing	Y (only [21])	Y(only [13])	N	Y
cooperation of multiple SDN controllers	N	N	N	Y
NFV in cloud	N	Y(only [24])	Y(only [28])	Y
Edge computing	N	N	Y(only [10], [35])	Y
NS in 5G and beyond networks	N	N	Y	Y
AI-based algorithm	N	Y(only [26], [37]	N	N

3. Proposed QoS Framework for Reliable NS in B5G System

In this section, we first introduce the proposed framework's design goals, and then describe the overall framework in detail. Then, we present the network resource model and resource allocation algorithms of NS adopted by our framework.

3.1 Framework Design Goals

After a deep investigation, we have determined that an ideal QoS framework for NS should ensure the features mentioned below. We have designed and developed our framework to achieve these features.

1. **Flexibility and reliability:**
   Flexibility and reliability are significant for the QoS framework of NS in a B5G system. Softwarization is one of the most important B5G enablers, in which VNFs can be highly customized according to the traffic needs. The NS should be customized to ensure the service level agreement (SLA) in terms of bandwidth, latency, jitter, and packet loss. Moreover, a smart life cycle management (LCM) of NS should be designed to meet the desired SLA. Similarly, the reliability of QoS refers to the stability needed to meet QoS requirements. For some latency-sensitive applications, such as autonomous driving and industrial manufacturing, network stability is crucial for ensuring the functionality of the system and human safety.
2. **QoS as a service for multi-tenants:**
   The second goal of the proposed framework is to provide QoS as a service (QaaS). Similar to the software as a service (SaaS) concept in cloud computing, QaaS is enabled by SDN and NFV technologies to change the B5G network industry's business model. In this case, multi-tenants can transparently use the same service provider's infrastructure while ensuring NS's isolation and SLA. Also, this concept provides the flexibility to apply the pay-as-you-go (PAYG) model of cloud computing, whereby customers pay only for what they have effectively used. The tenants can rent QoS from the main provider and the middle-providers, which work as an agent between the main provider and end users.
3. **Integrating both SDN and NFV technologies:**
   SDN provides a distributed architecture to control and manage all physical network resources, which is well-aligned with the QoS framework's scalability and flexibility. However, SDN can not effectively handle the life cycle of NS. Therefore, SDN and NFV must be integrated to manage virtualized resources and the life cycle of NS.

3.2 Framework Overview

Figure 1 shows the overall framework intended to achieve the design goals described above. The B5G network infrastructure is divided into three parts: radio access network (RAN), transport network (TN), and core network (CN). End users (e.g., mobile phones, webcams, smart industrial devices, vehicles, etc.) connect to B5G through the wireless base stations in RAN. In contrast to the previous generations, the B5G system is expected to host services close to the users by leveraging the multi-access edge computing (MEC) capability. In this case, many network functions are implemented as VNFs in MEC, enabling the softwarization of RAN. TN is located between the RAN and CN. Like traditional metropolitan area networks (MAN) and wide area networks (WAN), TN covers several kilometers or longer distances to connect different RANs and CNs. To enhance the TN's functions and management, the cloud data center is also built to sup-

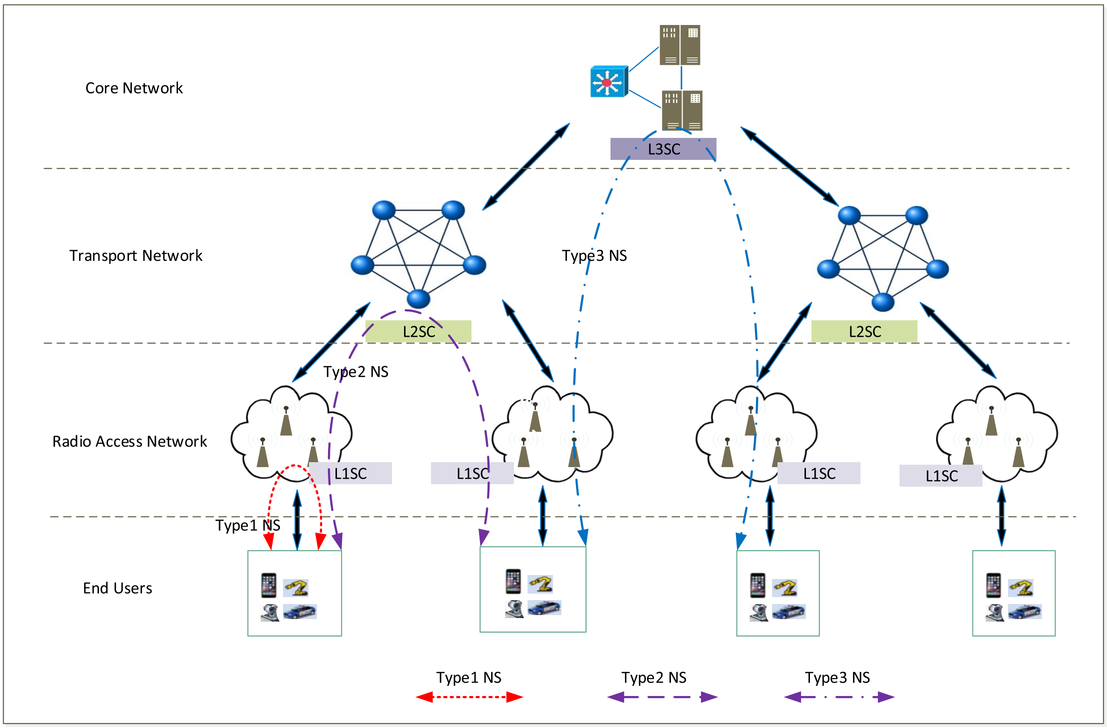


Fig. 1 Overall QoS framework in B5G based on SDN and NFV.

port SDN and NFV, which is responsible for WAN optical network management, mobile management, and user data analysis. CN is the whole management domain's schedule center, which orchestrates all VNFs or physical network functions (PNFs).

To manage the physical network resources, including network devices, computing and storage resources, a SDN and NFV based management system is designed in our framework. Each part of the physical network (RAN, TN, and CN) has its own management system that communicates to each other through its SDN controller. The SDN controller in RAN is called Level 1 SDN Controller (L1SC) in our framework. Similarly, the SDN controllers in TN and CN are called L2SC and L3SC, respectively. L1SC has the responsibility to report local network information to L2SC, while L2SC has also the responsibility to report local network information to L3SC. Finally, all L3SCs should synchronize all network information to keep the whole network consistent.

In this situation, each L1SC in RAN has a limited topology view bounded by one RAN domain. L1SC has the responsibility to communicate with the corresponding L2SC in TN without considering other L1SCs. Similarly, each L2SC in TN has a narrow topology view of one TN domain and reports its information to corresponding L3SC in CN without considering other L2SCs. For example, if two end users access to the same RAN, the communication between them is easily handled by the L1SC in this RAN. However, when end users are located in different RANs, L2SC will allocate related network resources, cooperating with two L1SCs. Furthermore, when end users are located across different TNs, L3SC may also take part in network resource allocation. Hence, the number of SDN controllers can be flexibly scaled up or down according to the network size.

Most industrial SDN controllers (e.g. ONOS and OpenDayLight) provide RESTful north-bound application programming interfaces. Different NSs with specific QoS requirements can be created based on these interfaces. According to the scope of network resources needed by NS, we define three types of NS as shown in Fig. 1. A Type 1 NS is created in a single RAN, implicating that the packet forwarding path from the source to the destination must be constrained in that RAN. A Type 2 NS is designed with a packet forwarding path such as RAN-TN-RAN, while a Type 3 NS is created with a packet forwarding path such as RAN-TN-CN-TN-RAN. Therefore, a Type 3 NS covers more network nodes and needs more network resources than a Type 1 NS and a Type 2 NS. When tenants apply for a network slice, they should consider both requirements, besides the cost, to decide which type of NS is the best choice.

3.3 Network Resource Model and Allocation Algorithm for Reliable NS

In this subsection, we describe the network resource model and mathematically formulate the proposed solutions.

1. **Formulation of network resources in 5G:** Let $G(V,E)$ represent a network topology managed by a SDN controller, in which V is the set of nodes and E is the set of links in this network. Let $b(i,j)$ and

$l(i,j)$ denote the available bandwidth and transmission latency between nodes i and j, respectively; $i,j \in V$ and $(i,j) \in E$ in network topology $G(V,E)$. For each $G(V,E)$ that contains n nodes, the following $n \times n$ matrix represents the available bandwidth between any two nodes.

$$B_G = \begin{bmatrix} b_{(1,1)} & b_{(1,2)} & \dots & b_{(1,n)} \\ b_{(2,1)} & b_{(2,2)} & \dots & b_{(2,n)} \\ \dots & \dots & \dots & \dots \\ b_{(n,1)} & b_{(n,2)} & \dots & b_{(n,n)} \end{bmatrix} \quad (1)$$

Since $b_{(i,j)}$ always equals $b_{(j,i)}$, B_G is a symmetric matrix. Furthermore, if two nodes that are not directly connected, we cannot obtain the bandwidth value exactly, because it is possible that there are multiple paths with different bandwidth values between them. Therefore, we assume that each element $b(i,j)$ of this matrix satisfies the following condition.

$$b_{(i,j)} = \begin{cases} bandwidth & if\,(i \neq j)\,\&\, connects\, directly \\ 0 & otherwise \end{cases} \quad (2)$$

Similarly, the following $n \times n$ symmetric matrix represents the transmission latency between any two nodes.

$$L_G = \begin{bmatrix} l_{(1,1)} & l_{(1,2)} & \dots & l_{(1,n)} \\ l_{(2,1)} & l_{(2,2)} & \dots & l_{(2,n)} \\ \dots & \dots & \dots & \dots \\ l_{(n,1)} & l_{(n,2)} & \dots & l_{(n,n)} \end{bmatrix} \quad (3)$$

where

$$l_{(i,j)} = \begin{cases} latency & if\,(i \neq j)\,\&\, connects\, directly \\ 0 & otherwise \end{cases} \quad (4)$$

We assume that the SDN controller can measure and collect all the data of B_G and L_G in its own management domain. However, according to our framework, multiple SDN controllers possibly coordinate to measure the QoS between different network management domains for Type 2 and Type 3 NS, respectively. For example, the forwarding path of a Type 2 NS looks like RAN-TN-RAN, so we should acquire the network status data between RAN and TN. In this vein, the following two $m \times n$ matrices represent the bandwidth and transmission latency between the network $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, respectively.

$$B_{G_1 \times G_2} = \begin{bmatrix} b_{(1,1)} & b_{(1,2)} & \dots & b_{(1,n)} \\ b_{(2,1)} & b_{(2,2)} & \dots & b_{(2,n)} \\ \dots & \dots & \dots & \dots \\ b_{(m,1)} & b_{(m,2)} & \dots & b_{(m,n)} \end{bmatrix} \quad (5)$$

$$L_{G_1 \times G_2} = \begin{bmatrix} l_{(1,1)} & l_{(1,2)} & \dots & l_{(1,n)} \\ l_{(2,1)} & l_{(2,2)} & \dots & l_{(2,n)} \\ \dots & \dots & \dots & \dots \\ l_{(m,1)} & l_{(m,2)} & \dots & l_{(m,n)} \end{bmatrix} \quad (6)$$

Similarly, when $i \in V_1$ does not connect directly to $j \in V_2$, $b_{(i,j)}$ and $l_{(i,j)}$ should equal to zero. Generally, G_1 and G_2 are far away from each other geographically so that there are very few direct links between them. Thus, it is clear that both $B_{G_1 \times G_2}$ and $L_{G_1 \times G_2}$ are large sparse matrices, since most of their elements are zero.

2. **E2E QoS requirements for reliable NS:**
We define $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ as the requirements of a QoS-sensitive NS. In this case, EU_s represents the source end users, while EU_d represents the destination end users. B_{min} and L_{max} denote the minimum bandwidth and the maximum latency of this NS, respectively. This means that the bandwidth between EU_s and EU_d should not be less than B_{min}, and maximum transmission latency between them should not be longer than L_{max}. For a Type 1 NS, both EU_s and EU_d connect with the nodes of the same RAN domain managed by a SDN controller, which is represented as $G_{RAN} = (V_{RAN}, E_{RAN})$. We assume that the SDN controller can detect two nodes $s \in V_{RAN}$ and $d \in V_{RAN}$ that connect EU_s and EU_d directly, respectively. If a specific forward path from s to d, described as $P_{(s \to d)} = s, v_1, v_2, \cdots, v_n, d$ can meet the QoS requirements of a Type 1 NS, then we have the following conditions:

$$\begin{aligned} B_{min} &\leqslant Min[b_{(s,v_1)}, b_{(v_1,v_2)}, \dots, b_{(v_{n-1},v_n)}, b_{(v_n,d)}] \\ L_{max} &\geqslant l_{(s,v_1)} + \sum_{i=1}^{n-1} l_{(v_i,v_{i+1})} + l_{(v_n,d)} \end{aligned} \quad (7)$$

For a Type 2 NS, the source and destination end users are located in different RANs that connect via the same TN. In this situation, the forward path looks like the following:

$$\begin{aligned} P_{(s \to d)} =& s, v_1^1, v_2^1, \dots, v_m^1, v_1^2, v_2^2, \dots, v_n^2, \\ & v_1^3, v_2^3, \dots, v_k^3, d \end{aligned}$$

where

$$\begin{aligned} v_i^1 (1 \leqslant i \leqslant m) &\in V_{RAN}^1 \\ v_i^2 (1 \leqslant i \leqslant n) &\in V_{TN}^2 \\ v_i^3 (1 \leqslant i \leqslant k) &\in V_{RAN}^3 \end{aligned}$$

The forwarding path for a Type 2 NS has the form RAN-TN-RAN and should meet the following QoS requirements.

$$\begin{aligned} B_{min} \leqslant & Min[b_{(s,v_1^1)}, b_{(v_1^1,v_2^1)}, \dots, \\ & b_{(v_{n-1}^1,v_n^1)}, b_{(v_n^1,v_1^2)}, b_{(v_1^2,v_2^2)}, \dots, \\ & b_{(v_{n-1}^2,v_n^2)}, b_{(v_n^2,v_1^3)}, b_{(v_1^3,v_2^3)}, \dots, \\ & b_{(v_{n-1}^3,v_n^3)}, b_{(v_n^3,d)})] \\ L_{max} \geqslant & l_{(s,v_1^1)} + \sum_{i=1}^{m-1} l_{(v_i^1,v_{i+1}^1)} + l_{(v_n^1,v_1^2)} + \sum_{i=1}^{m-1} l_{(v_i^2,v_{i+1}^2)} \end{aligned}$$

$$+ l_{(v_m^2,v_1^3)} + \sum_{i=1}^{k-1} l_{(v_i^3,v_{i+1}^3)} + l_{(v_n^3,d)}$$

Given the QoS requirements of a Type 2 NS, we can easily deduce the QoS requirements of a Type 3 NS. Obviously, it should look more complicated since a Type 3 NS has a longer forwarding path than a Type 2 NS, as described before.

3. **Allocation algorithms for reliable NS**
Allocating E2E network resources for different types of NS is a challenging task, which should leverage available network resources in the physical network and dynamic user requirements. To solve this problem, we propose corresponding network resource allocation algorithms for NS, which can effectively schedule network resources to satisfy the QoS requirements of different NS types. The core idea of the algorithms is to iterate Dijkstra algorithm to find multiple forwarding paths for a specific NS so that the flows of the same NS are forwarded along multiple paths to get to the destination. With this algorithm, SDN controller splits one big QoS requirement into several smaller ones. We first focus on the situation of a Type 1 NS, where all the end users connect to the same RAN. The input to this algorithm is the QoS requirements of a Type 1 NS, i.e. $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ and the available network resource information of the RAN, i.e. $B_{G_{RAN}(V_{RAN},E_{RAN})}$ and $L_{G_{RAN}(V_{RAN},E_{RAN})}$. We assume that the SDN controller has the ability to find the EU_s to connect the RAN through the node $s \in V_{RAN}$, and EU_d to connect the RAN thanks to the node $d \in V_{RAN}$. Therefore, the next key step is how to find one or more forwarding paths form s to d in the network topology of RAN to satisfy the QoS requirements B_{min} and L_{max} as shown in Expression (7). Algorithm 1 describes the process based on the mathematical formulation mentioned above.

In this algorithm, we supply four input parameters R_{NS},$B_{G_{RAN}}$,$L_{G_{RAN}}$ and Max_path_number. The first three parameters are self-explanatory, so we explain only the fourth one, Max_path_number. As described above, when a single forward path can not meet the requirements of a NS for some reason, such as a lack of sufficient bandwidth or too much transmission latency, we should try to find multiple paths that can cooperate to serve this NS. Theoretically, it is possible to find many paths unless the Dijkstra algorithm fails. However, we do not need too many forward paths for a single NS, because the more paths we have for a single NS, the more configuration we need to handle in the network, which probably increases the management cost and impacts the reliability of the whole network. In general, the optimal number of paths depends on the current network status. Therefore, we set a parameter Max_path_number that enables operators to flexibly decide the maximum number of paths in terms of the real network situation.

Following the initialization, Algorithm 1 gets into the while iteration where there is an important judgment statement, which is $B^t_{min} > b_{(v_x,v_{x+1})}$ && $L_{max} \geqslant L_{total}$. It deals with the situation that the path, generated by the Dijkstra algorithm, can not meet the bandwidth requirement but can meet the latency requirement of the NS. In this case, we will recompute the path using the Dijkstra algorithm to get a new path to satisfy the new bandwidth $B^t_{min} \leftarrow (B^t_{min} - b_{(v_x,v_{x+1})})$, in which $b_{(v_x,v_{x+1})}$ is the bandwidth that the previous path provides. Before recomputing the path using the Dijkstra algorithm, the bandwidth matrix $B_{G_{RAN}}$, denoted by matrix (1), should be updated by subtracting the value of $b_{(v_x,v_{x+1})}$ from all the related edge nodes of the previous path so that it may generate a different path compared to the previous one. Another important statement is $L_{max} < L_{total}$, which handles the situation, that the total latency of the path can not satisfy the requirement of the NS. The solution is to determine the edge nodes in the path which have maximum latency, and remove them from the network topology $G_{RAN}(V_{RAN}, E_{RAN})$, as well as to update $B_{G_{RAN}}$ and $L_{G_{RAN}}$, denoted by matrices (1) and (3), respectively. Then, the Dijkstra algorithm will probably generate different paths in the next iteration.

After the while iteration breaks, we should check two conditions $n == 0$ and $B_{min} \geqslant \sum_{i=1}^{n} b_{P^i}$. The first condition means that the Dijkstra algorithm failed to find any available path, and the second condition represents that it found one or more paths, but the bandwidth sum of these paths did not meet the requirements of the NS. Finally, if the whole algorithm successfully runs, it will return one or multiple paths and the corresponding bandwidth they can provide, which is represented by $\bigcup_{1\leqslant i\leqslant n}[P^i_{(s\to d)}, b_{P^i}]$. Otherwise, it will return empty set, which means it failed to find any available path.

For a Type 2 NS, we adopt a hop-by-hop policy to process the problem of network resource allocation based on Algorithm 1. Algorithm 2 describes this policy in detail. Algorithm 2 requires many more parameters than Algorithm 1 because the forwarding path takes the more complex form RAN-TN-RAN. In addition to the QoS requirements(R_{NS}) of the NS, we also feed the Algorithm 2 with all the bandwidth and latency data of two RANs and one TN, such as $B_{G^1_{RAN}}$, $B_{G_{TN}}$, $B_{G^2_{RAN}}$, $L_{G^1_{RAN}}$ and so on. The output of Algorithm 2 is the path sequence from EU_s to EU_d represented by $P^1_{RAN} \bigcup P_{TN} \bigcup P^2_{RAN}$. The core idea of Algorithm 2 is to consider each RAN or TN as a hop, where we use Algorithm 1 to compute and generate an internal path for each independently. If all internal paths can be generated successfully, they will be assembled to form a complete path sequence as the target output. Otherwise, it will return an empty set. Specifically, we first define two edge node sets that connect directly from RAN_1 to TN and from TN to RAN_2, denoted by $E_{(RAN_1\to TN)}$ and

$E_{(TN \rightarrow RAN_2)} = \Phi$, respectively. The two edge sets can be built from $B_{G^1_{RAN} \times G_{TN}}$ and $B_{G_{TN} \times G^2_{RAN}}$ using the filtering conditions $b_{(x,y)} \geqslant B_{min}$ and $l_{(x,y)} \leqslant L_{max}$. If both edge node sets are not empty, then we try to select $(x_1, y_1) \in E_{(RAN_1 \rightarrow TN)}$ and $(x_2, y_2) \in E_{TN \rightarrow RAN_2}$ to construct a full path $s \rightarrow x_1 \rightarrow y_1 \rightarrow x_2 \rightarrow y_2 \rightarrow d$. Through the $Foreach$ iteration, it will search all possible paths and find the target path to meet the QoS requirements.

The network resource allocation algorithm for a Type 3 NS is similar to that of a Type 2 NS, except that the former needs to handle more hops than the latter. It is easy to work out the algorithm for a Type 3 NS by extending Algorithm 2. For the sake of simplicity and to avoid repetition, we omit the description of the resource allocation algorithm for NS Type 3.

Algorithm 1: Network resource allocation for Type 1 NS.

Input:
$R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$
$B_{G_{RAN}(V_{RAN}, E_{RAN})}$
$L_{G_{RAN}(V_{RAN}, E_{RAN})}$
Max_path_number
Output:
$\bigcup_{1 \leqslant i \leqslant n} [P^i_{(s \rightarrow d)}, b_{Pi}]$ or Φ
1 Set $i = 1$ and $n = 0$;
2 Set $B^t_{min} = B_{min}$;
3 Get $s \in V_{RAN}$ and $d \in V_{RAN}$ from EU_s, EU_d;
4 **while** *true* **do**
5 Run Dijkstra algorithm with s,d and $B_{G_{RAN}}$
6 Get a shortest path $P^i_{(s \rightarrow d)}$ from s to d;
7 **if** $P^i_{(s \rightarrow d)} \neq \Phi$ && $i \leqslant Max\_path\_number$ **then**
8   $b_{(v_x, v_{x+1})} \leftarrow Min[b_{(s,v_1)}, b_{(v_1,v_2)}, \ldots,$
9   $b_{(v_{n-1},v_n)}, b_{(v_n,d)}]$ in $P^i_{(s \rightarrow d)}$;
10   $L_{total} \leftarrow l_{(s,v_1)} + \sum_{j=1}^{n-1} l_{(v_j,v_{j+1})} + l_{(v_n,d)}$;
11 **else**
12   $n \leftarrow (i - 1)$;
13   goto 20;
14 **end**
15 **if** $B^t_{min} \leqslant b_{(v_x,v_{x+1})}$ && $L_{max} \geqslant L_{total}$ **then**
16   $b_{Pi} \leftarrow b_{(v_x,v_{x+1})}$;
17   $n \leftarrow i$;
18   goto 20;
19 **else**
20   **if** $B^t_{min} > b_{(v_x,v_{x+1})}$ && $L_{max} \geqslant L_{total}$ **then**
21    $b_{(v_x,v_{x+1})} \leftarrow Min[b_{(s,v_1)}, b_{(v_1,v_2)}, \ldots,$
22    $b_{(v_{n-1},v_n)}, b_{(v_n,d)}]$ in $P^i_{(s \rightarrow d)}$;
23    $B^t_{min} \leftarrow (B^t_{min} - b_{(v_x,v_{x+1})})$;
24   Update $B_{G_{RAN}}$ by subtracting the $b_{(v_x,v_{x+1})}$ along the path $P^i_{(s \rightarrow d)}$;
25    $b_{Pi} \leftarrow b_{(v_x,v_{x+1})}$;
26    $i \leftarrow (i + 1)$;
27   goto 5;
28   **end**
29   **if** $L_{max} < L_{total}$ **then**
30    $l_{(v_x,v_{x+1})} \leftarrow Max[l_{(s,v_1)}, l_{(v_1,v_2)}, \ldots,$
31    $l_{(v_{n-1},v_n)}, l_{(v_n,d)}]$ in $P^i_{(s \rightarrow d)}$;
32   Remove edge $(v_x, v_{x+1}) \in E_{RAN}$
33   from $G_{RAN}(V_{RAN}, E_{RAN})$;
34   Update $B_{G_{RAN}}$ and $L_{G_{RAN}}$;
35   goto 5;
36   **end**
37 **end**
38 **end**
39 **if** $n == 0 \,||\, B_{min} \geqslant \sum_{i=1}^{n} b_{Pi}$ **then**
40 Return Φ;
41 **else**
42 Return $\bigcup_{1 \leqslant i \leqslant n} [P^i_{(s \rightarrow d)}, b_{Pi}]$;
43 **end**

Algorithm 2: Network resource allocation for a Type2 NS.

Input:
$R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$
$B_{G^1_{RAN}}, B_{G_{TN}}, B_{G^2_{RAN}}$
$B_{G^1_{RAN} \times G_{TN}}, B_{G_{TN} \times G^2_{RAN}}$
$L_{G^1_{RAN}}, L_{G_{TN}}, L_{G^2_{RAN}}$
$L_{G^1_{RAN} \times G_{TN}}, L_{G_{TN} \times G^2_{RAN}}$
Output:
$P^1_{RAN} \bigcup P_{TN} \bigcup P^2_{RAN}$ or Φ
1 Set $E_{(RAN_1 \rightarrow TN)} = \Phi$;
2 Set $E_{(TN \rightarrow RAN_2)} = \Phi$;
3 Get $s \in V^1_{RAN}$ and $d \in V^2_{RAN}$ from EU_s, EU_d;
foreach $b_{(x,y)} \in B_{G^1_{RAN} \times G_{TN}}$ &&$b_{(x,y)} \neq 0$ **do**
4 **if** $b_{(x,y)} \geqslant B_{min}$&&$l_{(x,y)} \leqslant L_{max}$ **then**
5   $E_{(RAN_1 \rightarrow TN)} \leftarrow E_{(RAN_1 \rightarrow TN)} \bigcup (x, y)$;
6 **end**
7 **end**
8 **foreach** $b_{(x,y)} \neq 0$&&$b_{(x,y)} \in B_{G_{TN} \times G^2_{RAN}}$ **do**
9 **if** $b_{(x,y)} \geqslant B_{min}$&&$l_{(x,y)} \leqslant L_{max}$ **then**
10   $E_{(TN \rightarrow RAN_2)} \leftarrow E_{(TN \rightarrow RAN_2)} \bigcup (x, y)$;
11 **end**
12 **end**
13 **if** $E_{(RAN_1 \rightarrow TN)} = \Phi$ $||$ $E_{(TN \rightarrow RAN_2)} = \Phi$ **then**
14 Return Φ;
15 **end**
16 **foreach** $(x_1, y_1) \in E_{(RAN_1 \rightarrow TN)}$ **do**
17 Run Algorithm 1, Get P^1_{RAN} from s to x_1;
18 **foreach** $(x_2, y_2) \in E_{TN \rightarrow RAN_2}$ **do**
19   Run Algorithm 1, get P_{TN} from y_1 to x_2 Run Algorithm 1, get P^2_{RAN} from y_2 to d
   $L_{total} \leftarrow L_{P^1_{RAN}} + L_{P_{TN}} + L_{P^2_{RAN}}$;
20   **if** $L_{total} \leqslant L_{max}$ **then**
21   Return $P^1_{RAN} \bigcup P_{TN} \bigcup P^2_{RAN}$;
22   **end**
23 **end**
24 Return Φ;
25 **end**

4. Experimental Evaluation

In this section, we first introduce the experimental environment and then evaluate the performance of the algorithms described in Sect. 3.

4.1 Experiment Environment Setup

We conduct the experiments on two computers, each with an Intel multi-core i5-4300 CPU and 8 Gb RAM. The two computers are connected directly through network interfaces with a speed of 1G/bps. We select ONOS as our experimental SDN controller, Mininet as the network topology emulation tool and OVS as the virtual switch in the network nodes. ONOS runs on one computer and Mininet/OVS runs on another computer.

ONOS is a highly-modular distributed controller, which can be deployed in a large-scale network to form a cluster of controllers. Each controller can manage its own part of the network nodes, communicating with other controllers to keep the consistency of the real-time network status. In addition, ONOS provides many flexible north-bound RESTful interfaces to allow users to develop a new application module, 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 ONOS application module which implements the following functions:

- Collect the bandwidth and latency data between any two nodes that connect directly.
- Compute the shortest path according to QoS requirements based on Algorithm 1 and Algorithm 2 for various packet flows. Generally, the packet flow is recognized by the pair of source IP address and destination IP address, so we take it as a NS in this context.
- Allocate bandwidth for a specific flow along the shortest path, which can be handled by ONOS remotely based on the queue feature of the OVS port.
- Steer a specific flow into a specific queue of the OVS port through modifying the flow table in the OVS. Each queue can be configured with a minimum and maximum bandwidth. When a flow is steered into a specific queue, the transmission speed of this flow will be constrained between the minimum and maximum bandwidth configured for this queue, which is guaranteed by the packet schedule algorithm in OVS, such as HTB (Hierarchy Token Bucket) or HFSC (Hierarchical Fair Service Curve). For example, let's assume there are two ports $p1$ and $p2$ in an OVS node and two flows $f1$ and $f2$ belonging to two different NS, respectively. If both $f1$ and $f2$ pass through this OVS from $p1$ to $p2$, we can create two queues $q1$ and $q2$ for port $p2$. Then, $f1$ and $f2$ can be steered into $q1$ and $q2$ when they enter this OVS from $p1$ and leave from $p2$. Consequently, the transmission speed of $f1$ or $f2$ should not be less than the minimum bandwidth and should not exceed the maximum bandwidth of the corresponding queue. In this way, ONOS and Mininet/OVS can be used to simulate the network performance when different types of NS are created in various network environments.

4.2 Performance Evaluation

This evaluation assesses the processing time of the algorithms and the bandwidth utilization of the networks after every NS request. The processing time is particularly important for users when they request a new NS with specific requirements. However, the processing time may be related with many factors, such as the number of network nodes, current network state, the position of end users, the computing efficiency of the underlying algorithm, etc. In the beginning, there is enough available bandwidth in the network, so the algorithms can find a forwarding path for a new NS very quickly. As the available bandwidth and the number of links in the network decrease, the algorithms will intuitively take a longer time to find suitable paths, and may even fail to find one. In addition, the size and type of network topology also affect the execution time of the algorithms to some extent. Therefore, by investigating the processing time for creating a new NS and the bandwidth utilization of networks, we can observe the performance of the proposed algorithms for different types of NS and different network statuses, which may provide some useful guidance in practice. In order to observe the relationship between the processing time and the related factors, we measure the processing time and the bandwidth utilization of networks by sending 100 creation requests for new NS continuously in the same network topology, which is designed as a full-mesh network. Any two nodes in a full-mesh network connect directly and the reason why we choose full-mesh network is to better observe the behavior of the algorithms for different NS types.

We first test the situation of requesting new Type 1 NS in a full-mesh network that contains 20 and 100 OVS nodes. The bandwidth and latency matrix of the network is generated randomly with the constraints $b_{(i,j)} \in [50\,Mbps, 100\,Mbps]$ and $l_{(i,j)} \in [1\,ms, 10\,ms]$. Fig. 2 shows the network topology and experiment environment for Algorithm 1.

For each Type 1 NS request, the QoS requirements $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ are also generated ran-

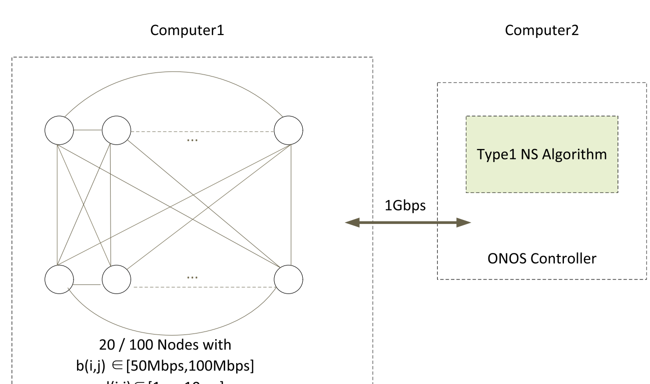


Fig. 2 Network topology and experiment environment for evaluating the performance of Algorithm 1.

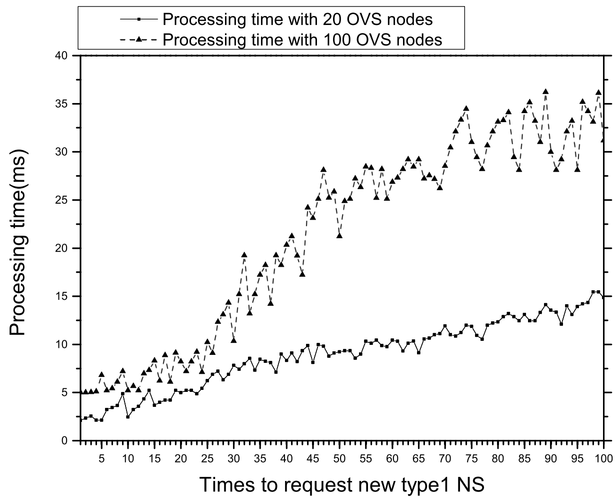


Fig. 3 The processing time for accommodating new requests for Type 1 network slices with constraints of $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$.

domly, which means that the positions of the end users are changing randomly. As for the minimum bandwidth and maximum latency, we have the constraints of $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$. Figure 3 presents the processing time for the Type 1 NS request based on the above situation. In the figure, the Y-axis denotes the processing time in milliseconds and the X-axis shows the 100 requests for the creation of a new Type 1 NS, numbered from 1 to 100. The straight line represents the processing time in the full-mesh network with 20 OVS nodes, while the triangular dotted line shows the processing time in case of the network with 100 OVS nodes. For the sake of convenience, we call the full-mesh network with 20 OVS nodes as Network A and the full-mesh network with 100 OVS nodes as Network B. We can see that as the number of requests for new NS increases, the processing time of Network A increases from 2.5 ms to 15 ms at an approximately linear rate. This shows that the available network resources in the same network are decreasing, and the SDN controller takes 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 observe that the processing time for Network B increases faster than that for Network A. As expected, the reason is that the target path in Network B becomes more complicated than that of Network A with the decrease of network resources in it. Moreover, there is a slight fluctuation in both lines, which means that sometimes a new request requires less time than the previous one. For example, the processing time of the 74^{th} request in Network B is about 35 ms while that of the 77^{th} request is about 28 ms. We believe this may be caused by the random positions of the end users as input parameters of Algorithm 1.

Next, we measure the bandwidth utilization of the underlying network. The bandwidth utilization of a network is defined as the mean value of the bandwidth utilization of all the links in the network. Figure 4 shows the bandwidth utilization after creating each of the 100 envisioned Type 1 NS in case of a RAN (i.e., as we focus on Type 1 NS) consisting of 20 nodes and 100 nodes, respectively. As expected, the bandwidth utilization of RAN increases as the number of requests for new NS grows. From the figure, it becomes also clear that the bandwidth utilization of 20 nodes increases more quickly than that of 100 nodes. At the end, after the 100th request of a new NS, the bandwidth utilization of 20 nodes RAN gets to about 60 percent, while the bandwidth utilization of 100 nodes RAN gets to only about 38 percent. The reason is that the more nodes a network has, the more paths a new NS can select from in this network. Therefore, the usage of links becomes more balanced and more bandwidth remains available.

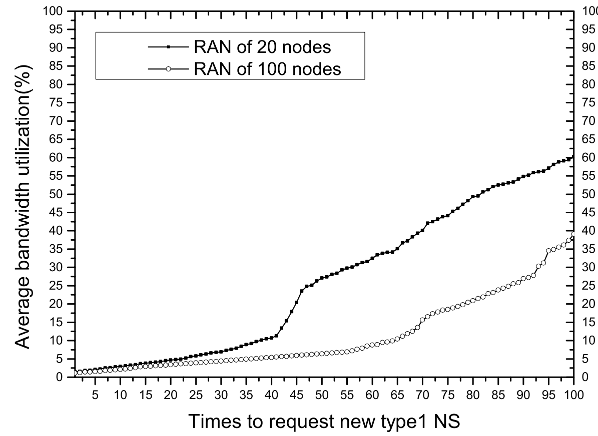


Fig. 4 The bandwidth utilization of the network in case of Algorithm1.

Similarly to the evaluation method for Type 1 NS, we conduct experiments of requesting new Type 2 NS in a more complicated network topology that contains three full-mesh networks, which represent RAN1, TN, and RAN2, respectively. RAN1 and TN are connected only by two links, which mimics the long-distance fiber connection between two network domains in a real network environment. Similarly, TN and RAN2 are also connected by two links directly. We set the bandwidth and latency of the links between two full-mesh networks as 200 $Mbps$ and 1 ms. The bandwidth and latency matrices of the three full-mesh networks are generated randomly in the same way used in the Type 1 NS. Figure 5 shows the network topology for evaluating the performance of Algorithm 2.

For each Type 2 NS request, the QoS requirements $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ are also generated randomly with the constraints $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$. Figure 6 presents the relationship between the processing time and the number of requests for Type 2 NS based on this situation.

In Fig. 6, 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. Similarly, each network domain of Network B contains 100 OVS nodes. From the figure, we observe that

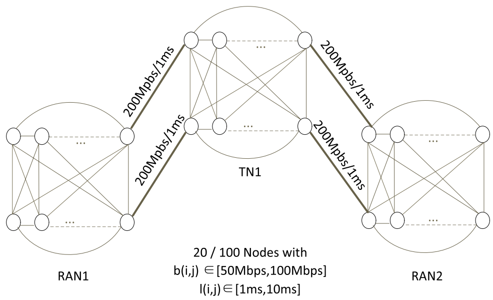


Fig. 5 The network topology for evaluating the performance of Algorithm 2.

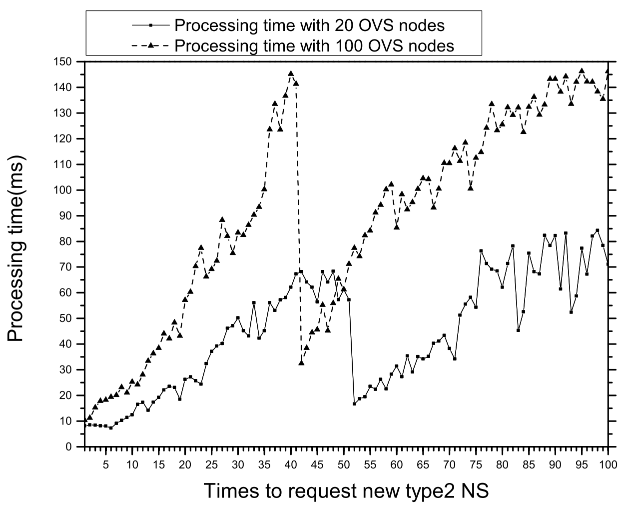


Fig. 6 The processing time for accommodating new requests for Type 2 network slices with constraints of $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$.

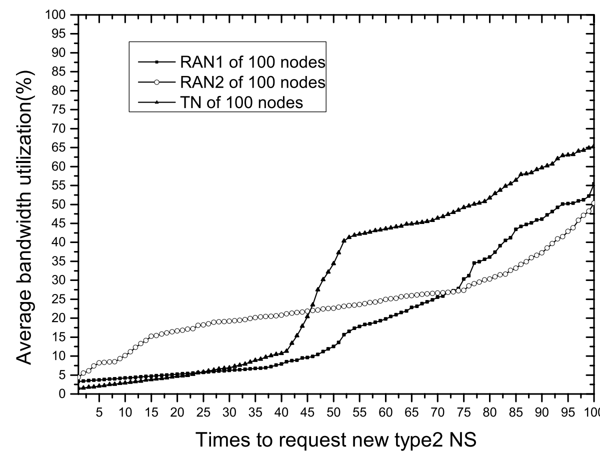


Fig. 7 The network bandwidth utilization in case of Algorithm 2.

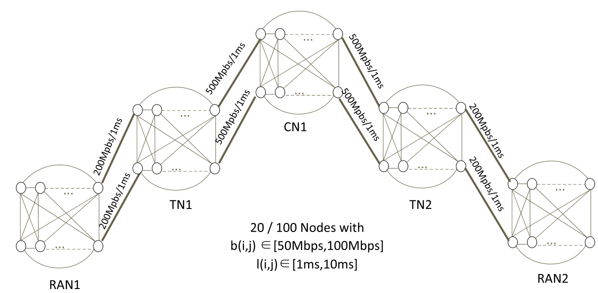


Fig. 8 The network topology for evaluating the performance of Algorithm 3.

the processing time for Type 2 NS experiences a sudden decrease when the number of requests for new NS is between 40 and 50. For Network A, the processing time decreases from about 60 ms to 18 ms at the 50^{th} request. For Network B, the processing time decreases from about 140 ms to 25 ms at the 40^{th} request. The main reason for this relates to the policy of the search path in Algorithm 2. When the bandwidth of one link between two network domains is consumed, Algorithm 2 will choose another link to find an E2E path. In this case, the edge nodes for the two network domains are changed, which probably produces a shorter path inside each network domain and finally reduces the processing time. We can also make a quantitative analysis. The bandwidth of one link between two network domains is 200 $Mbps$, and the constraint of NS requirements is $B_{min} \in [1\,Mbps, 5\,Mbps]$, therefore it is reasonable to consume all the bandwidth of one link when the number of requests is about 40 to 50.

Next, we measure the bandwidth utilization of Network B when applying Algorithm 2. Figure 7 shows the bandwidth utilization of RAN1, RAN2 and TN, respectively after creating each of the 100 Type 2 NS. In Fig. 7, we observe that the bandwidth utilization of RAN1 and RAN2 is similar as the number of Type 2 NS requests increases, but the bandwidth utilization of TN increases more quickly than that of RAN1 and RAN2. At the end, after the 100^{th} request for a new NS, the bandwidth utilization of TN gets to about 65 percent, while the bandwidth utilization of RAN1 and RAN2 reaches only about 50 percent and 55 percent, respectively. We believe that the reason for this is that all the edge nodes of TN are fixed when the new NS is created, so the NS paths in the TN are located intensively, which will probably lead to higher bandwidth utilization.

To validate Algorithm 3 for Type 3 NS, we construct a more complicated network topology. Figure 8 shows the network topology for evaluating the performance of Algorithm 3. As shown in Fig. 8, the network topology contains five full-mesh networks, including RAN1, RAN2, TN1, TN2, and CN. RAN1 and TN1, RAN2, and TN2 are each connected only by two direct links, and we set the bandwidth and latency of these links to 200 $Mbps$ and 1 ms. TN1 and CN, and TN2 and CN are also connected by two links, and we set the bandwidth and latency of these links to 500 $Mbps$ and 1 ms. Similarly, the bandwidth and latency matrices of the five full-mesh networks are generated randomly in the same way used in the Type 2 NS, and every full-mesh network contains 20 or 100 OVS nodes.

For each Type 3 NS request, the QoS requirements $R_{NS} = (EU_s, EU_d, B_{min}, L_{max})$ are also generated randomly with the constrains $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$. Figure 9 presents the relationship between

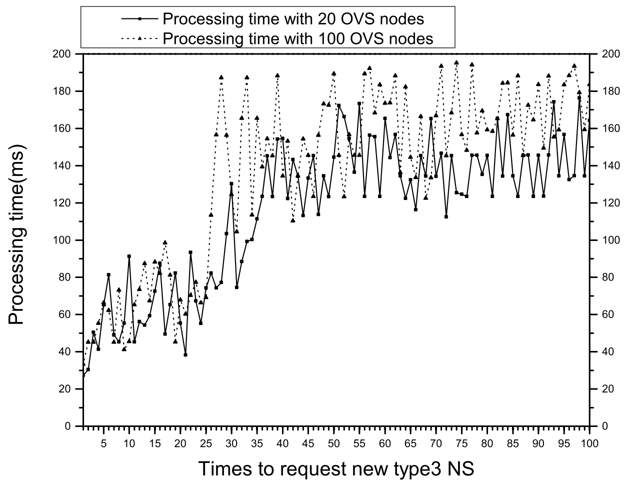


Fig. 9 The processing time for accommodating requests for Type 3 network slices with constraints of $B_{min} \in [1\,Mbps, 5\,Mbps]$ and $L_{max} \in [50\,ms, 100\,ms]$.

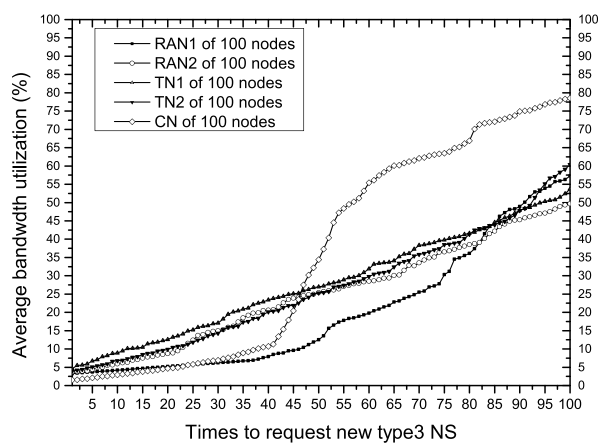


Fig. 10 The network bandwidth utilization in case of Algorithm 3.

the processing time and the number of requests for Type 3 NS based on this situation.

In Fig. 9, similarly, we refer to the network with 20 OVS nodes in each of its domains as Network A, and the network with 100 OVS nodes in each of its domains as Network B. We observe that the processing time for Type 3 NS experiences a sudden rise when the number of requests is between 25 and 30. For Network A, the processing time increases from about 80 ms to 130 ms at the 28^{th} request. For Network B, the processing time increases from about 80 ms to 180 ms at the 30^{th} request. We believe the main reason for this relates to the network topology for Algorithm 3. When the number of network domains is greater than three and the number of new NS requests reaches a certain threshold, the number of iterations of Algorithm 3 will sharply increase, which consequently consumes more processing time. Another experimental result is that the processing times of new Type 3 NS in case of Network A and B are very similar, which indicates that the scale of nodes in the network is not an important factor for the processing time due to the high complexity of Algorithm 3.

In the same way, we measure the bandwidth utilization in case of Network B (i.e., consisting of 100 nodes in each domain) for Algorithm 3. Figure 10 shows the bandwidth utilization of RAN1, RAN2, TN1, TN2, and CN, respectively, after creating each of the 100 Type 3 NS. In Fig. 10, we notice that the bandwidth utilizations of RAN1 and RAN2 are similar as the number of Type 3 NS requests increases, but the bandwidth utilizations of TN1 and TN2 are slightly higher than those of RAN1 and RAN2. The bandwidth utilization of CN is the highest among all the network domains. At the end, after the 100^{th} request of Type 3 NS, the bandwidth utilization of CN gets to about 78 percent, while the bandwidth utilization of the other network domains reaches between 50 percent and 60 percent. The reason for this is that all the edge nodes of CN are fixed. When a new NS is created, the NS paths in the TN or CN are located intensively, which will probably lead to higher bandwidth utilization. Therefore, in a real network, we should improve the structure of the network topology to eliminate these problems. For example, we can simplify the structure of TN and CN, or increase the bandwidth of their inner links.

5. Conclusion

In this paper, we proposed a QoS framework for fine-granular network slicing in Beyond 5G systems that is based on SDN and NFV. By dividing the network into three parts, namely RAN, TN, and CN, we described the function modules of the QoS framework in each domain. We then defined a requirement model of the B5G network resources and presented three different types of NS. Following the formulation of the network resources in NS, algorithms for network resource allocation for different NSs were proposed. Finally, we conducted simulations to evaluate the performance of these algorithms. The results showed that the proposed algorithms could satisfy the QoS requirements of different end users and may provide useful guidance to create and deploy NS in B5G. Future work will involve applying the algorithms in a real network and improving the algorithms according to the obtained feedback.

Acknowledgments

This research work was supported in part by the European Union's Horizon 2020 Research and Innovation Program under the Cloud for Holography and Augmented Reality (CHARITY) Project under Agreement 101016509, and in part by the Academy of Finland Project 6Genesis under Agreement 318927, and in part by the Fujian Province Natural Science Foundation of China under Agreement 2020J01574. Prof. Song was supported by IITP grant funded by the Korea government (No.2018-0-01456, AutoMaTa).

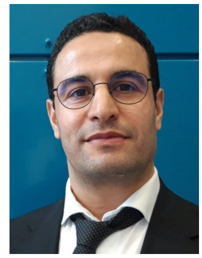
References

[1] P. Popovski, K. Trillingsgaard, O. Simeone, and G. Durisi, "5G

wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," IEEE Access, vol.6, pp.55765–55779, 2018.
[2] "A blueprint of technology, applications and market drivers towards the year 2030 and beyond," ITU-T FG-NET-2030, ITU, Geneva, Switzerland, May 2019.
[3] M.H. Alsharif, A.H. Kelechi, M.A. Albreem, S.A. Chaudhry, M.S. Zia, and S. Kim, "Sixth generation (6G) wireless networks: Vision, research activities, challenges and potential solutions," Symmetry, vol.12, no.4, 2020.
[4] W. Jiang, B. Han, M.A. Habibi, and H.D. Schotten, "The road towards 6G: A comprehensive survey," IEEE Open Journal of the Communications Society, vol.2, pp.334–366, 2021.
[5] S. Zhang, "An overview of network slicing for 5G," IEEE Wireless Commun., vol.26, no.3, pp.111–117, 2019.
[6] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," IEEE Commun. Surveys Tuts., vol.20, no.3, pp.2429–2453, 2018.
[7] 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 Commun. Mag., vol.55, no.5, pp.80–87, 2017.
[8] M.S. Bonfim, K.L. Dias, and S.F. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," ACM Comput. Surv. (CSUR), vol.51, no.6, p.114, 2019.
[9] H. Zhang, N. Liu, X. Chu, K. Long, A.H. Aghvami, and V.C. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," IEEE Commun. Mag., vol.55, no.8, pp.138–145, 2017.
[10] A.N. Toosi, R. Mahmud, Q. Chi, and R. Buyya, "Management and orchestration of network slices in 5G, fog, edge and clouds," Fog and Edge Computing: Principles and Paradigms, pp.79–101, 2019.
[11] D.A. Chekired, M.A. Togou, L. Khoukhi, and A. Ksentini, "5G-slicing-enabled scalable SDN core network: Toward an ultra-low latency of autonomous driving service," IEEE J. Sel. Areas Commun., vol.37, no.8, pp.1769–1782, 2019.
[12] C. Bektas, S. Böcker, F. Kurtz, and C. Wietfeld, "Reliable software-defined RAN network slicing for mission-critical 5G communication networks," 2019 IEEE Globecom Workshops (GC Wkshps), 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] Q. Wang, J. Alcaraz-Calero, R. Ricart-Sanchez, M.B. Weiss, A. Gavras, N. Nikaein, X. Vasilakos, B. Giacomo, G. Pietro, M. Roddy, M. Healy, P. Walsh, T. Truong, Z. Bozakov, K. Koutsopoulos, P. Neves, C. Patachia-Sultanoiu, M. Iordache, E. Oproiu, I.G.B. Yahia, C. Angelo, C. Zotti, G. Celozzi, D. Morris, R. Figueiredo, D. Lorenz, S. Spadaro, G. Agapiou, A. Aleixo, and C. Lomba, "Enable advanced QoS-aware network slicing in 5G networks for slice-based media use cases," IEEE Trans. Broadcast., vol.65, no.2, pp.444–453, 2019.
[15] S. Do, L.V. Le, B.S.P. Lin, and L.P. Tung, "SDN/NFV based internet of things for multi-tenant networks," Trans. Netw. Commun., vol.6, no.6, p.40, 2019.
[16] J.W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation," IEEE Commun. Surveys Tuts., vol.20, no.1, pp.388–415, 2018.
[17] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," IEEE Netw., vol.33, no.6, pp.172–179, 2019.
[18] Z. Shu and T. Taleb, "A novel QoS framework for network slicing in 5G and beyond networks based on SDN and NFV," IEEE Netw., vol.34, no.3, pp.256–263, 2020.
[19] S. Tomovic, N. Prasad, and I. Radusinovic, "SDN control framework for QoS provisioning," 2014 22nd Telecommunications Forum Telfor (TELFOR), pp.111–114, IEEE, 2014.
[20] M. Bagaa, D.L.C. Dutra, T. Taleb, and K. Samdanis, "On SDN-driven network optimization and QoS aware routing using multiple paths," IEEE Trans. Wireless Commun., vol.19, no.7, pp.4700–4714, 2020.
[21] D.L.C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," GLOBECOM 2017-2017 IEEE Global Communications Conference, pp.1–6, IEEE, 2017.
[22] T. Pan, T. Huang, J. Mao, C. Li, and Y. Liu, "OpenSched: Programmable packet queuing and scheduling for centralized QoS control," 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp.95–96, IEEE, 2017.
[23] 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," 2018 IEEE Symposium on Computers and Communications (ISCC), pp.00602–00607, IEEE, 2018.
[24] 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.
[25] M. Rezaee and M.H.Y. Moghaddam, "SDN-based quality of service networking for wide area measurement system," IEEE Trans. Ind. Inf., vol.16, no.5, pp.3018–3028, 2020.
[26] 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.
[27] L. Tello-Oquendo, S.C. Lin, I.F. Akyildiz, and V. Pla, "Software-defined architecture for QoS-aware iot deployments in 5G systems," Ad Hoc Networks, vol.93, p.101911, 2019.
[28] R. Montero, A. Pagès, F. Agraz, and S. Spadaro, "Supporting QoE/QoS-aware end-to-end network slicing in future 5G-enabled optical networks," Metro and Data Center Optical Networks and Short-Reach Links II, p.109460F, International Society for Optics and Photonics, 2019.
[29] A. Sgambelluri, O. Dugeon, A. Muhammad, J. Martín-Pérez, F. Ubaldi, K. Sevilla, O. De Dios, T. Pepe, C. Bernardos, P. Monti, and F. Paolucci, "Orchestrating QoS-based connectivity services in a multi-operator sandbox," J. Opt. Commun. Netw., vol.11, no.2, pp.A196–A208, 2019.
[30] Q. Ye, J. Li, K. Qu, W. Zhuang, X.S. Shen, and X. Li, "End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework," IEEE Veh. Technol. Mag., vol.13, no.2, pp.65–74, 2018.
[31] L. Baldini, Q. Wang, J.A. Calero, M.B. Weiss, A. Gavras, G. Bernini, P.G. Giardina, C. Angelo, X. Vasilakos, C.-Y. Chang, N. Nikaein, S. Spadaro, A. Pages, F. Agraz, G. Agapiou, T. Truong, K. Koutsopoulos, J. Cabaca, and R. Figueiredo, "Slicenet control plane for 5G network slicing in evolving future networks," 2019 IEEE Conference on Network Softwarization (NetSoft), pp.450–457, IEEE, 2019.
[32] R. Ricart-Sanchez, P. Malagon, A. Matencio-Escolar, J.M. Alcaraz Calero, and Q. Wang, "Toward hardware-accelerated QoS-aware 5G network slicing based on data plane programmability," Trans. Emerging Telecommunications Technologies, vol.31, no.1, p.e3726, 2019.
[33] B. Han, J. Lianghai, and H.D. Schotten, "Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks," IEEE Access, vol.6, pp.33137–33147, 2018.
[34] X. Feng, Z. Lu, L. Wang, and W. Guan, "A delay-aware deployment policy for end-to-end 5G network slicing," ICC 2019-2019 IEEE International Conference on Communications (ICC), pp.1–6, IEEE, 2019.
[35] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5G networks," ICC 2019-2019 IEEE International Conference on Communications (ICC), pp.1–7, IEEE, 2019.

[36] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," China Commun., vol.15, no.10, pp.86–98, 2018.
[37] Y. Zhou, Z.M. Fadlullah, B. Mao, and N. Kato, "A deep-learning-based radio resource assignment technique for 5G ultra dense networks," IEEE Netw., vol.32, no.6, pp.28–34, 2018.
[38] Y. Fu, S. Wang, C.X. Wang, X. Hong, and S. McLaughlin, "Artificial intelligence to manage network traffic of 5G wireless networks," IEEE Netw., vol.32, no.6, pp.58–64, 2018.

Zhaogang Shu is currently an Associate Professor at the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, China. He also is the director of the department of computer science and Cloud Computing Lab, Fujian Agriculture and Forestry University. He received B.S. and M.S. degrees in computer science from Shantou University, China in 2002 and 2005 respectively. He also received Ph.D. degree from South China University of Technology, Guangzhou, China, in 2008. From Sept. 2008 to July 2012, he worked as a senior engineer and project manager at Ruijie Network Corporation, Fuzhou, China. From Oct. 2018 to Oct. 2019, he worked as a visiting professor at the department of communications and networking, School of Electrical Engineering, Aalto University, Finland. His research interests include software-defined network, network function virtualization, 5G network, network security, machine learning and cloud computing. He serves as the reviewers of many famous journals on network technology, including IEEE Network, IEEE/ACM Transactions on Networking, ACM/Springer Mobile Networks and Applications. He directed more than 8 research projects and was the author of 30 papers and 5 patents.



Tarik Taleb is currently a Professor at the Center of Wireless Communications, The University of Oulu, Finland. He is the founder and director of the MOSA!C Lab (www.mosaic-lab.org). Between Oct. 2014 and Dec. 2021, he has been a Professor at the School of Electrical Engineering, Aalto University, Finland. Prior to that, he was working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. He was then leading the NEC Europe Labs Team working on R&D projects on carrier cloud platforms. Before joining NEC and till Mar. 2009, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI, the second largest mobile operator in Japan. From Oct. 2005 till Mar. 2006, he worked as research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B.E. degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively. Prof. Taleb's research interests lie in the field of telco cloud, network softwarization & slicing, AI-based software defined security, immersive communications, & next generation mobile networking. Prof. Taleb has been also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP's System Architecture working group 2. Prof. Taleb is a member of the IEEE Communications Society Standardization Program Development Board. As an attempt to bridge the gap between academia and industry, Prof. Taleb founded the "IEEE Workshop on Telecommuni-cations Standards: from Research to Standards", a successful event that got awarded "best workshop award" by IEEE Communication Society (Com-SoC). Based on the success of this workshop, Prof. Taleb has also founded and has been the steering committee chair of the IEEE Conf. on Standards for Communications and Networking. Prof. Taleb served as the general chair of the 2019 edition of the IEEE Wireless Communications and Net-working Conference (WCNC'19) held in Marrakech, Morocco. He was the guest editor in chief of the IEEE JSAC Series on Network Softwarization & Enablers. He is/was on the editorial board of the IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine, IEEE Journal on Internet of Things, IEEE Transactions on Vehicular Technology, IEEE Communications Surveys & Tutorials, and a number of Wiley journals. Till Dec. 2016, he served as chair of the Wireless Communications Technical Committee, the largest in IEEE ComSoC. He also served as Vice Chair of the Satellite and Space Communications Technical Committee of IEEE ComSoc (2006–2010). Prof. Taleb is the recipient of the 2021 IEEE ComSoc Wireless Communications Technical Committee Recognition Award (Dec. 2021), the 2017 IEEE ComSoc Communications Software Technical Achievement Award (Dec. 2017) for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize (May 2017), the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher award (Jun. 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Founda-tion (Mar. 2008), the 2007 Funai Foundation Science Promotion Award (Apr. 2007), the 2006 IEEE Computer Society Japan Chapter Young Au-thor Award (Dec. 2006), the Niwa Yasujirou Memorial Award (Feb. 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003). Some of Prof. Taleb's research work have been also awarded best paper awards at prestigious IEEE-flagged conferences.

Jaeseung Song is currently an Associate Professor, leading the Software Engineering and Security Lab (SESLab), in the Computer and Information Security Department at Sejong University. He received the B.S. and M.S. degrees in computer science from Sogang University, and the Ph.D. degree from the Department of Computing, Imperial College London, U.K. He worked for NEC Europe Ltd., from 2012 and 2013, as a leading Standard Senior Researcher. From 2002 to 2008, he worked for LG Electronics as a Senior Researcher, leading a 3GPP SA Standard Team. His research interests span the areas of smart health, wearable IoT devices and platforms, software engineering, software testing, networked systems and security, with a focus on the design and engineering of reliable IoT/M2M platforms, particularly in the context of IoT technologies for pandemic, secure software patch techniques, the blockchain IoT, and edge computing. He holds the position of the IEEE ComSoc Industry Communities Board IoT Coordinator, oneM2M Technical Plenary Vice-Chair, and the TTA IoT/M2M Convergence Special Project Group Chair. He is an Associate Editor of IEEE IoT Journal and an Editor of IoT series in IEEE Communications Standards Magazine. He has been general chair and TPC Chair of many international conferences, including IEEE CSCN, IEEE WF-IoT, IEEE ICC, and IEEE GLOBECOM. He has been a guest editor in IEEE Communications Magazine and Sensors.