Journal of Information Processing Vol.25 153–163 (Feb. 2017)

[DOI: 10.2197/ipsjjip.25.153]

Invited Paper

End-to-end Network Slicing for 5G Mobile Networks

Akihiro Nakao[1,a)] Ping Du[1,b)] Yoshiaki Kiriha[1,c)] Fabrizio Granelli[2,d)]
Anteneh Atumo Gebremariam[2,e)] Tarik Taleb[3,f)] Miloud Bagaa[3,g)]

Received: July 5, 2016, Accepted: December 7, 2016

Abstract: The research and development (R&D) and the standardization of the 5th Generation (5G) mobile networking technologies are proceeding at a rapid pace all around the world. In this paper, we introduce the emerging concept of network slicing that is considered one of the most significant technology challenges for 5G mobile networking infrastructure, summarize our preliminary research efforts to enable end-to-end network slicing for 5G mobile networking, and finally discuss application use cases that should drive the designs of the infrastructure of network slicing.

Keywords: network slicing, 5G mobile network, Internet-of-Things (IoT)

1. Introduction

The research and development (R&D) of the 5th Generation (5G) mobile networking technologies is making rapid progress around the globe. For example, there are fora formed all around the world, such as METIS [34], NGMN [38], 5GPPP [3] in Europe, 4G/5G Americas [23] in U.S., 5G Forum [2] in Korea, and IMT-2020 [7] in China.

In Japan, the Fifth Generation Mobile Networking Promotion Forum (5GMF) was established in September, 2014 and published its white paper in 2016 [41]. 5GMF network architecture committee defines the key concepts in 5G mobile networks as supporting extremely flexible communication infrastructure to maintain end-to-end quality of applications of different requirements, as well as identifies four fundamental technical requirements, network softwarization and slicing, mobile edge computing (MEC), mobile front-haul and back-haul technologies, network management and orchestration.

In the meantime, in the standards developing organizations (SDOs) arena for network (non-radio) technologies for 5G mobile networks, ITU-T [8] has formed the focus group (FG) IMT-2020 in 2015, to conduct gap analysis among missing technologies to be enabled to realize 5G mobile networks. The 5GMF network architecture committee has been contributing much to the gap analysis and has led the discussions especially on advanced technical concepts such as network softwarization and network slicing, and information centric networking in mobile networking [17]. Also, 3GPP [1] has rapidly accelerated standardization for LTE-Advanced and 5G mobile networks, including network technologies such as slicing, QoS, etc. as work items for next generation networks.

In the light of these trends around network slicing technologies, we observe that the development of these advanced technologies are driven by emerging applications in mobile networks that have a wide spectrum of different requirements for network, computational, and storage resources. We strongly believe that the concept of network slicing be one of the most significant technologies among all. Network slicing, we believe, in a nutshell, is an extension of the concept of "slice," which has been often used in SDN, NFV, distributed cloud research as an isolated set of programmable resources to program network functions to deal with various application requirements, to mobile networks, especially discussed in 5G mobile network context.

Although the research and development on network slicing are still in its infancy, in this paper, we introduce the emerging concept of network slicing that is considered one of the most significant technology challenges for 5G mobile network infrastructure, summarize our preliminary research efforts to enable end-to-end network slicing for 5G mobile networks, and finally discuss the application use cases that should drive the designs of the infrastructure of network slicing.

The rest of the paper is structured as follows. Section 2 introduces the emerging concept of end-to-end network slicing. Sections 3 and 4 show our prototyping efforts on packet core slicing and RAN slicing. Section 5 discusses slicing structure. Section 6 discusses application use cases we are trying to accommodate in different slices to demonstrate the benefit of end-to-end network slicing. Finally, Section 7 briefly concludes.

2. End-to-end Network Slicing

2.1 Network Slicing

The terminology "network slicing" has recently attracted much attention in industries and SDOs such as 3GPP [1] and ITU [8]. Although the definition of network slicing is still under heavy

[1] The University of Tokyo, Bunkyo, Tokyo 113–0033, Japan
[2] University of Trento, via Calepina, 14–38122 Trento, Italy
[3] Aalto University, Otakaari 5, 02150 Espoo, Finland
[a)] nakao@iii.u-tokyo.ac.jp
[b)] duping@iii.u-tokyo.ac.jp
[c)] ykiriha@iii.u-tokyo.ac.jp
[d)] fabrizio.granelli@unitn.it
[e)] anteneh.gebremariam@unitn.it
[f)] tarik.taleb@aalto.fi
[g)] miloud.bagaa@aalto.fi

Table 1 5G use case examples and their QoS requirements.

5G Use Cases	Examples	Requirements	Mobility
eMBB/xMBB	4K/8K ultra high definition (UHD) video, hologram, Augmented Reality (AR), Virtual Reality (VR)	High capacity, video cache	Yes
mMTC	Sensor Networks (smart metering, logistics, city, home, etc.)	Massive connection covering a very large area of mostly immobile devices	No
URLLC/uMTC	Autonomous driving, smart-grid, remote surgery	Low latency and high reliability	Yes

discussion, we have generally defined "slice" as *an isolated set of programmable resources to implement network functions and application services through software programs* for accommodating individual network functions and application services within each slice without interfering with the other functions and services on the coexisting slices [36].

Network slicing is considered one of the most important concepts to realize "extreme flexibility" in 5G mobile networks [41]. The current mobile networks are optimized to serve only mobile phones. However, 5G mobile networks need to serve a variety of devices with very different, heterogeneous quality of service (QoS) requirements without interference among one another.

The 5GMF white paper [41] classifies communications to be enabled in 5G mobile networks into three categories, eMBB (enhanced Mobile Broad Band), mMTC (massive Machine Type Communications), and URLLC (Ultra Reliable and Low Latency Communications). Also, the 5G infrastructure public private partnership (5G PPP) defines three use cases, xMBB (massive broad band) to deliver gigabytes of bandwidth to mobile devices on demand, mMTC also known as massive Internet of Things (IoT) targeted to connect immobile sensors and machines, and uMTC (ultra-reliable Machine-Type Communications) to allow immediate feedback with high reliability in cases like autonomous driving, remote controlled robots. Although the terminologies are different, these three categories of communications defined in two organizations as shown in **Table 1** imply that 5G mobile networks must support very different QoS for each type of communications, thus address the requirements for supporting extreme flexibility in 5G mobile networks.

2.2 A Brief History of Network Slicing

The concept of slice in networking has been first introduced in the overlay network research efforts, such as PlanetLab [13], in 2002. At that time, a slice has been defined as an isolated set of network bandwidth, computational, storage, resources allocated for a group of users that "program" network functions and services over their overlay networks overlaid on top of the planet. Since then, various network virtualization testbed research efforts such as PlanetLab EU [14], GENI [12], VNode [48], FLARE [6], Fed4Fire [5], have inherited the concept of slices as a basis of the infrastructures.

Considering the concept of slicing developed initially for (fixed) network research testbeds to accommodate various different types of services and applications sharing the same physical infrastructures, when we face similar demands in applications of 5G mobile networks, we may quite naturally extend the slicing concept to mobile networks, including transport, packet core, access networks and wireless radio networks. Network slicing for 5G mobile network is thus largely considered an extension of the concept of "slice" that has been originally defined and developed since early 2000s, to recent mobile networking R&D with additional focus on mobile network functions implemented on top of programmable resources.

2.3 Network Softwarization

Network softwarization is an overall transformation trend for designing, implementing, deploying, managing and maintaining network equipment and network components by software programming, exploiting characteristics of software such as flexibility and rapidity of design, development and deployment throughout the lifecycle of network equipment and components [16], [17].

Network softwarization is another driving factor for network slicing, as realization of network functions and equipment by software program is considered a key to the network slicing concept. Software-defined networking (SDN) and network function virtualization (NFV) are the two main enabling technologies in creating network slices on physical infrastructure resources.

2.4 End-to-end Network Slicing for 5G Mobile Networks

As discussed above, the concept of network slicing has originated from overlay network and network virtualization research efforts. Although the concept of network slicing for mobile networks is relatively new in the technology world today, it is rapidly gaining momentum as we proceed into the future.

5GMF network architecture committee led by the authors of this paper has identified the importance of end-to-end network slicing, all the way from user equipment (UE) to cloud data centers for enabling end-to-end quality and extreme flexibility to accommodate various applications, as shown in **Fig. 1** included in the white paper [41].

Figure 1 captures two significant aspects of end-to-end network slicing. First, down at the bottom, we have physical infrastructure, that is, a collection of network, computing and storage resources, all the way along end-to-end communication. We should be able to use these programmable resources embedded along the end-to-end communication paths.

Second, we should be able to logically define an end-to-end network slice from UE to cloud data centers using the programmable resources per application service. This means that we need to enable dynamic creation, modification, and destroy of network slice, coordinated especially across fixed network and

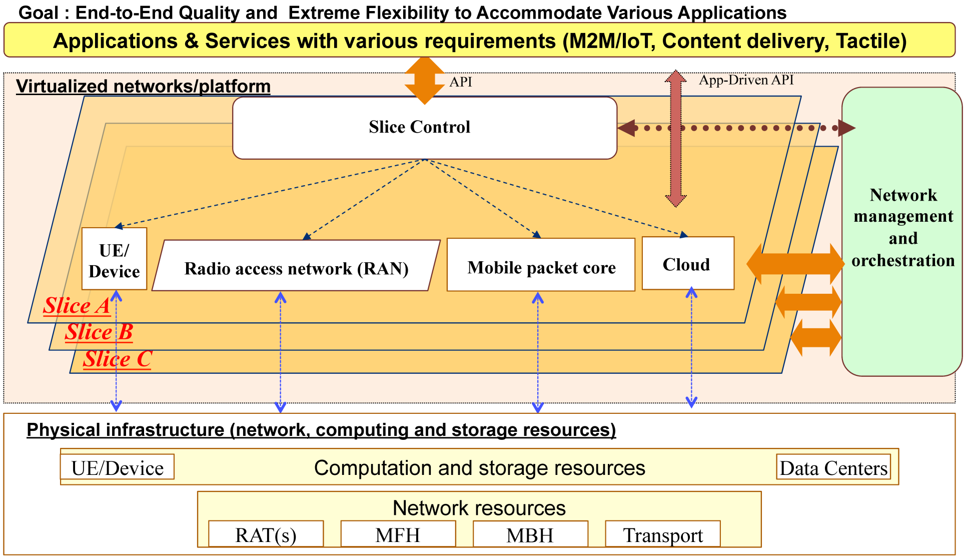
Goal : End-to-End Quality and Extreme Flexibility to Accommodate Various Applications
Applications & Services with various requirements (M2M/IoT, Content delivery, Tactile)
Virtualized networks/platform
API
App-Driven API
Slice Control
Network management and orchestration
UE/ Device
Radio access network (RAN)
Mobile packet core
Cloud
Slice A
Slice B
Slice C
Physical infrastructure (network, computing and storage resources)
UE/Device
Computation and storage resources
Data Centers
Network resources
RAT(s)
MFH
MBH
Transport

Fig. 1 End-to-end slicing concept [41].

radio boundary, so called mobile packet core slicing and RAN (Radio Access Network) slicing. Each network slice is made up of a virtualized air-interface, radio access network and mobile packet core network, and transport network combined. We also note that mobile fronthaul and backhaul network slicing need to be considered as well.

3. Packet Core Slicing

There are several research efforts to slice packet core networks using virtual machines on top of white boxes with general purpose processors and network processors as well as FPGAs. In this section, we first review the components of packet core network and introduce our preliminary research efforts on slicing packet core networks using deeply programmable network nodes with general purpose processors and network processors, called FLARE programmable nodes [6] and FPGA boards with OpenAirInterface (OAI) [39] on top of them.

3.1 Packet Core Network

In 3GPP, the current generation Evolvable Packet Core (EPC) network has been designed and standardized as a flat architecture, where IP (Internet Protocol) is the only protocol to transport all services. A User Equipment (UE) can get Internet connectivity when connected to EPC over RAN (radio access network). In considering end-to-end network slicing, we must fist study packet core network slicing as an extension to well studied transport network slicing, such as transport SDN.

3.2 MVNO as Precursor to Slicing

Mobile Virtual Network Operators (MVNOs) that obtain sliced RAN resources from Mobile Network Operators (MNOs) act as a forerunner in mobile network slicing, which is a key technology in the evolution towards 5G cellular wireless networks. Comparing to the traditional MNOs, MVNOs don’t need to build their own RAN but focus on building new business models and new value-added network functions in their packet core networks.

We believe that an in-depth study of MVNO network not only help the current MVNOs improve their services but also benefit the 5G research and development in terms of network slicing, especially for allocating isolated resources for different applications and services.

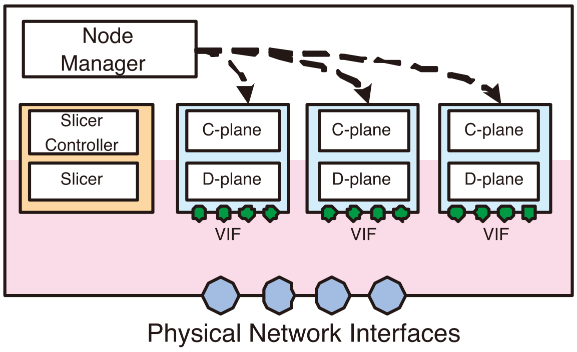

Fig. 2 Underlying infrastructure of FLARE [6].

For example, in Refs. [25], [37], we create application specific slices and characterize the flow properties of popular mobile applications. We have studied how to create per-application slices within an MVNO using deeply programmable FLARE [6] nodes. Since the data plane of our MVNO is enabled to be deeply programmable by FLARE nodes, from given flows observed at the data plane, we can identify applications that transmit those flows with 100% accuracy. We also develop an application-specific in-network processing mechanism to optimize MVNO network through applying different virtual network functions to different applications.

3.3 FLARE Programmable Node

FLARE [6] is an open deeply programmable node architecture that can concurrently run multiple isolated virtual network functions on a physical node. As shown in **Fig. 2**, a FLARE node consists of a combination of many-core network processors and Intel x86 processors. With various virtualization techniques, we slice both resources (CPUs, memory and link bandwidth) in both many-core processors and x86 processors. We define *sliver* as a node portion of a slice allocated over the entire network. From here on, we use slice and slivers interchangeably, but the scope of a slice is network wide while that of a sliver is within a single node. For each sliver, control plane runs on x86 processors while data plane runs on many-core processors. Control plane and data plane communicate via Ethernet-over-PCIe interface.

All incoming packets are scanned and classified by *Slicer* and then classified to slivers. Although not shown in the figure, there is a central node called *FLARE central* that talks to the control modules called *Node Manager* to manage the resources of each FLARE node. *Node Manger* is in charge of adding/removing slivers at a FLARE node. Users can also configure and program their slivers via the interface provided by *FLARE central*.

3.4 Slicing eNB on FLARE

OpenAirInterface (OAI) [39] is an open experimentation and prototyping platform created by EURECOM. It provides a software implementation of all elements of the 4G LTE/5G architecture including user equipment (UE), eNodeB (eNB), Home Subscriber Server (HSS) and Evolved Packet Core (EPC) components. A compound EPC component consists of Serving Gateway (S-GW), Packet Data Network Gateway (P-GW) as well as the Mobility Management Entity (MME). The eNB and EPC components are responsible for creating channels (namely bearers) with UE and forwarding the user traffic.

As shown in **Fig. 3**, we run an eNB instance in a Docker [33] instance inside a FLARE sliver. Docker is a lightweight virtualization technology where each Docker instance only consists of the minimal running environment of each application. Each eNB in a Docker instance is isolated and replaceable within a FLARE sliver.

In our prototype, UEs connect to an eNB instance via software radio platform USRP B210 [20], which connects to a FLARE sliver via USB pass-through technology. The traffic flows of different slices are isolated using VLANs on the FLARE node running Open vSwitch [10]. The packets from different eNBs are classified and tagged with different *VLAN IDs* and then diverted to an EPC instance.

3.5 Slicing EPC on FLARE

Performance is an important factor to be considered in programmable 5G networks, which is highly dependent on the underlying hardware infrastructure. Hardware EPC appliances can achieve high performance but may lack flexibility in changing their functions once the logic has been programmed.

In EASE [45], Tarik et al. has introduced a Virtual Machine (VM)-based EPC slicing approach, where both control plane and data plane of an EPC slice are implemented in VMs. VM-based EPC slice running on top of commodity servers is completely flexible, but its performance is still suboptimal compared to purpose-built hardware devices especially when a large amount of data traffic needs to be classified and forwarded to VMs via hypervisor and processed there. To balance the flexibility and performance, we choose many-core network processors as the platform to prototype the data-path of the EPC slice.

In this section, we introduce how to implement an EPC slice in a FLARE slice shown in **Fig. 4**, where signaling related EPC entities (e.g., MME) are implemented in control plane while user data forwarding and processing (e.g., S-GW and P-GW) are implemented in data plane. Compared with EASE, one benefit of our approach is to reduce the user data processing delay at an EPC slice as well as increasing computing and processing capability through many-core processors.

3.5.1 Data Plane

We offload the GTP-U channel creation and user data processing from control plane to data plane, which is implemented with GTPV1-U kernel module in naive OAI software. One challenge of EPC implementation lies in offering such extensibility while at the same time achieving good performance. In order to scale network functions, one promising approach is to subdivide and modularize functionalities and to parallelize packet processing across on-chip multiple processors.

FLARE nodes enable rapid deployment of new network functions by providing Click network-programming framework [24], [31]. We abstract the underlying architecture such as I/O engine, inter-core communication and only expose the relevant necessary details to a set of predefined Click elements.

We implement SP-GW data plane components with chained Click elements. When a FLARE node receives packets from eNB, its classifier called *Slice slicer* will classify packets to different slices as well as classifying signaling packets, e.g., GTP-C from data packets, e.g., GTP-U. The signaling packets are forwarded to control plane while the data packets are processed in data plane with many-core processors.

3.5.2 Control Plane

We run the signaling entities of an EPC slice, e.g., MME and the control plane of SP-GW, in a Docker instance. We can

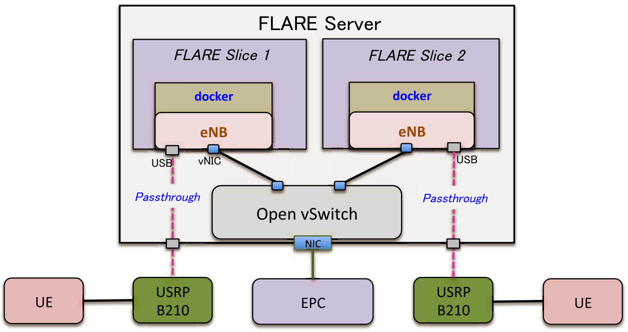

Fig. 3 Slicing eNB on FLARE server.

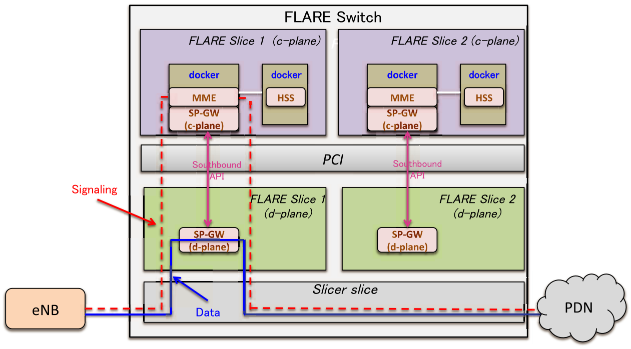

Fig. 4 Slicing EPC on FLARE switch.

also run HSS entity in another Docker instance within the same FLARE sliver. These two docker instances are isolated and replaceable without interfering with each other. For example, we can install different versions of packages in MME and HSS instances even if they may conflict with each other when installed on the same host machine.

The interfaces between EPC and HSS entities are implemented with internal Ethernet links. They can communicate with each other via TCP and SCTP protocols.

3.5.3 Southbound API

We need to define the Southbound API between data plane and control plane so that a *GTP-U* tunnel from data plane to eNB can be established when parsing and processing *GTP-C* packets in control plane.

When receiving *GTP-C* packets, MME asks SP-GW to establish, update and maintain the *GTP-U* tunnels in data plane. It is also responsible for transferring GTP tunneling parameters including endpoint identifiers with the Tunnel Endpoint Identifiers (TEIDs) to eNBs.

In implementing our prototype, we follow OpenFlow's convention for programming abstraction. We define our own programming abstraction as API as follows,

<UEID, TEID><Action><Stat>,

where UEID may be a UE's IP address assigned by MME through the signaling channel and Action may be actions such as create/update/remove a *GTP-U* tunnel.

4. RAN Slicing

4.1 Slicing Radio Access Network

Leveraging the SDN concept, each network slice can be tailored to a particular communication service in such a way that the control plane (CP) and user plane (UP) are configured according to different QoS requirements, while defining an open interface between them. In this section, we focus our discussions on Radio Access Network (RAN) slicing, where the most relevant operations that could be tailored to a particular network slice include mobile association, access control, load balancing and resource scheduling. In order to achieve those functionalities, the authors in Ref. [30] present different kinds of CP/UP configurations in a network slice with their corresponding cost of implementation and flexibility comparisons; i) common CP across slices and dedicated UP for each of the slices, ii) dedicated control/user planes and iii) common CP and dedicated UP for each slice. Choosing which configuration to mainly consider depends on the tradeoff between flexibility and cost of implementation.

Once we separate the CP and UP of the RAN, a virtualization technique is employed to create the virtual instances of the considered RAN resources. Depending on the selected CP and UP configurations, each virtual RAN instance contains a dedicated UP and a common CP or a dedicated CP/UP or a common CP with dedicated UP.

The amount of RAN resources allocated per network slice depends on the QoS requirements, e.g., as defined in Table 1), the amount of traffic load, the wireless link quality, etc. Moreover, in presence of multiple radio access technologies (RATs), it is also possible to allocate a RAN resource to a network slice from a single or multiple RATs. In order to improve flexibility, depending on the network traffic load, for example to dynamically turn on/off a particular RAT in a slice reduces the energy consumption of a network slice.

Considering a 4G long-term evolution (LTE) protocol stack, CP and UP of the Uu (between user equipment-UE and evolved Node B-eNB) and S1-interfaces (between eNB and EPC) can be separated to provide slice-specific configuration. In the Uu-interface, the UP consists of the packet data convergence protocol (PDCP), radio link control (RLC) and medium access control (MAC) layers in the protocol stack whereas the CP in addition to PDCP, RLC and MAC layers it also includes the radio resource controller (RRC). However, the main focus of this analysis is on RAN slicing, i.e., at the air-interface, as we describe it in the following subsection.

4.2 RAN Slicing on OpenAirInterface

In order to emulate RAN slicing at the Uu-interface, we consider an OAI [39] software/hardware testbed on top of an LTE technology. **Figure 5** depicts the protocol stack of the OAI soft UE, soft eNB and soft EPC. The ExpressMIMO2 board [39] is used for real time experimentation and validation of the communication between UEs and eNBs. Instead of ExpressMIMO2 UEs, commercial-off-the-shelf (COTS) UEs such as smartphones and LTE dongles can also be used. Intel x86-based PCs with Linux operating system installed are used to emulate the EPC as shown in Section 3.

Based on the experimentation platform defined in Fig. 5, we

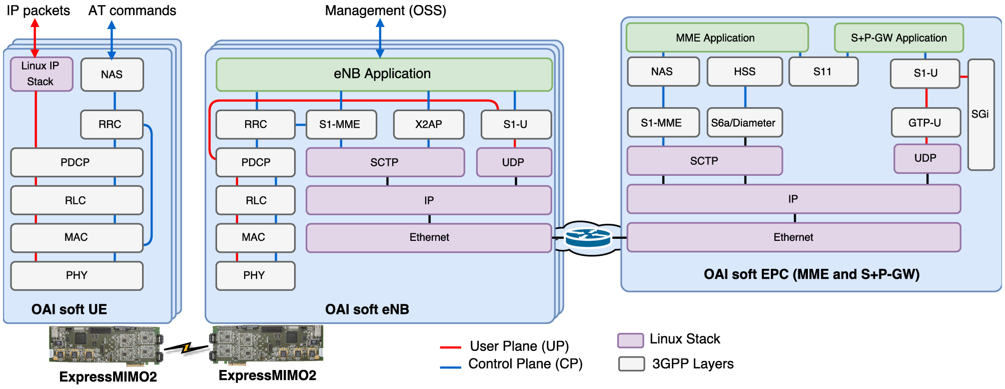

Fig. 5 OAI LTE protocol stack and ExpressMIMO2 hardware [39].

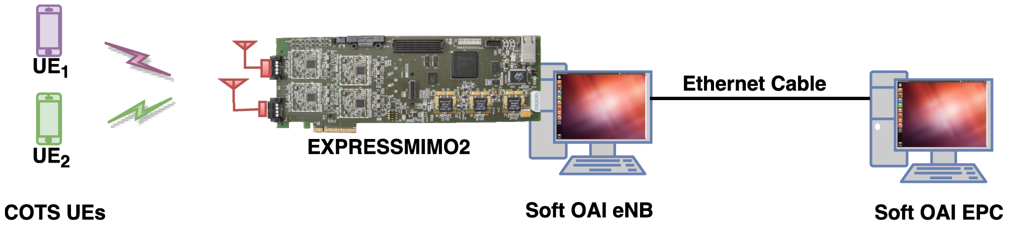

Fig. 6 OAI based prototype to implement RAN radio resource slicing.

propose a dynamic radio resource slicing, also known as dynamic spectrum-level slicing (DSLS) [22], technique using a single eNB and a single EPC. The prototype system model is shown in **Fig. 6** where we use a single ExpressMIMO2 board to emulate eNB (i.e., Virtual eNB (VeNB) instances are created by partitioning the radio resources an eNB into multiple parts each belonging to each of the VeNB instances), 2 COTS UEs and one Intel-based PC to run the EPC. The DSLS algorithm dynamically partitions the radio resources of the eNB into two portions each serving VeNB1 and VeNB2 respectively. The detailed problem formulation and algorithm implementation of the DSLS can be referred in Ref. [22].

Since the current version of OAI soft eNB only supports 2 UEs, we model the user distribution in each VeNB as the amount of traffic load generated by each of the UEs, UE1 and UE2, as depicted in Fig. 6, where the radio resources allocated to VeNB1 and VeNB2 serve UE1 and UE2 respectively.

4.3 RAN Slicing for IoT

Internet of Things (IoT) devices require low power wide area wireless technologies as these devices often run without rich power sources. From the point of view of 3GPP, LTE Cat-1 (10/5 Mbps, 20 MHz, Rel-8 [15]), LTE Cat-0 (375k or 1M/375k or 1 Mbps, 20 MHz, Rel-12 [18]), LTE Cat-M1 (300k or 800k/300k or 1 Mbps, 1.4 MHz, Rel-13 [19]) and Cat-NB1 (21/62 kbps, 200 kHz, Rel-13 [19]) are promising low power LTE technology family to be deployed. RAN slicing makes a lot of sense especially when we plan to use the same physical (softwarized) infrastructure for supporting various types of RATs.

We also note that many unlicensed wireless technologies such as Wi-Fi, WiGig [29], and LPWA (Low Power Wide Area) [47] like LoRa [9] and SigFox [11] are catching much attention in industries when it comes to 5G mobile networking.

We strongly believe that the pressure for accommodating these multiple RATs, licensed or unlicensed, in RAN slices is getting larger and larger as we have a proliferating number of devices and sensors for IoT industries. We plan to work on RAN slicing for accommodating multiple RATs in immediate future.

5. Hierarchical/Recursive Network Slicing

In Section 3, we show that we can create multiple slices of LTE network using FLARE platforms. This is one way of achieving end-to-end network slicing. However, RAN slicing introduced in Section 3 allows us to create subslices within a slice created to accommodate a single LTE network, say, per device, per application, etc.

This means that we can define hierarchical, in other words, recursive slicing structure, according to operational demands, combining these two techniques, one shown in Section 3 and the other in Section 4.

For example, an LTE network slice may be allocated per operator, while within the slice, subslices may be allocated for individual applications and/or devices. For another instance, an LTE network slice and a 5G network slice can coexist on top of a single physical infrastructure, while within each slice, subslices may be instantiated for each network instance. In the latter scenario, the migration from LTE to 5G may be feasible. One can imagine more interesting use cases other than these.

Through these observations, we believe that end-to-end network slicing may be composed of hierarchical/recursive structures in 5G mobile networking. To this end, we anticipate many interesting challenges ahead of us, and also we plan to learn solutions from combining together our two prototypes, packet core slicing through FLARE platforms and RAN slicing through OAI.

6. Use Cases

In this section, we introduce three representative use case scenarios of ultra-reliable low latency communications (URLLC), mMTC, and eMBB that have very different resource requirements, as discussed in Section 2. We envision that our system with network slicing enabled accommodate all these three different types of communications in different slices so that they do not interfere with one another. Although currently we are jointly working on these application use cases independently of our network slicing prototype system, we plan to combine our infrastructure with these application use cases to show the benefits of end-to-end network slicing.

6.1 URLLC: Smart Drive-assistant Services

As one of vertical industries emerging in 5G mobile networking, Intelligent Transportation System (ITS) research is now attacking URLLC applications, namely, smart drive-assistant services and cooperative driving services. In order to realize safe, energy-efficient, stress-free driving environment, as well as to optimize overall energy consumption in our societies and to reduce air pollution, coordinated control of a large number of cars from cloud and edge computing is considered promising.

As an early R&D result by the authors in the University of Tokyo, an infrastructure-based vehicle control system has already proposed by introducing coordinated control from edge and cloud servers. In Ref. [43], the University of Tokyo proposes a system architecture and control mechanisms and validates the utility of the proposal for maintaining highly stable control of cars in an area, while optimizing the utility of computational resources at cloud and edge servers. We expect that computing resources at edges be much more expensive than those in cloud data centers, as we must deploy such computational resources in wider areas to cover geographically and also because distributed systems generally incur high cost of operation and maintenance.

There are lots of research efforts on *autonomous driving* in ITS by newly introduced machine learning and Artificial Intelligence (AI) techniques. Only autonomous control of cars, however, may not completely realize smart driving services, since the overall optimization of energy consumption and the reduction of air solutions require simultaneous control of multiple vehicles taking into account global, i.e., wide area optimization as well as local, i.e., each vehicle information. We posit that we need to consider *collaborative driving*, where not only local information is considered, wider area information needs to be collected, summarized, and processed to optimize the overall ITS because in order to improve the correctness of a decision, accuracy of recognition, timeliness of control, and reliability of services, more frequent interactions and collaborations among vehicles, various sensors are necessary.

For realizing collaborative driving, we must be able to allocate a slice where we have low latency and ultra reliable communication to the edge servers with resources along the end-to-end communication paths. We believe that we need more points of edge computations than just a single edge cloud within the network as is often discussed in today's mobile edge computing, as depending on how much geographical coverage is required for optimization, we must exercise computations farther from UEs.

6.2 mMTC: Advanced Telemetric Probing

The evolving 5G cellular wireless networks are envisioned to not only interconnect smartphones and tablets, but also support massive Machine Type Communication (mMTC) including cars, drones, industrial machines and sensors, and provide mobile services with high data rate, low latency, and low energy consumption.

In order to study the use case of mMTC, we have been jointly operating an MVNO network for transmitting IoT traffic [26] in the University of Tokyo. We deploy over 400 Intel Edison [4] IoT gateways in operating public transportation vehicles such as buses, which frequently update GPS location, accelerometers, gyros, etc. by sending the data collected from the sensors to the central controller in the MVNO network.

Figure 7 shows the traffic pattern observed at the packet gateway (built on top of FLARE) of the MVNO network. We observed that the traffic volume decreases periodically, which shows that TCP global synchronization [44] persists for the TCP connections from a large number of IoT gateways since they have similar traffic characteristics, e.g., rate and RTT, and reduce their windows at the same time under congestion. Due to the global synchronization, the air bandwidth is not effectively used in mMTC since statically only about half of the air bandwidth is in use.

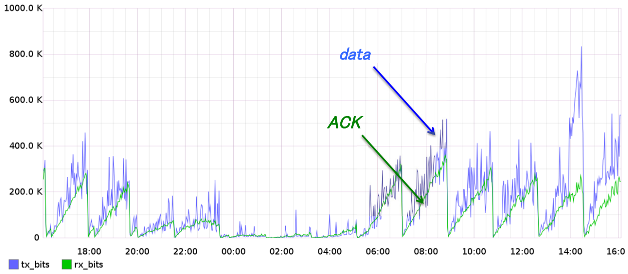

Fig. 7 An example of mMTC traffic pattern from massive IoT gateways.

To avoid such kind of global synchronization in mMTC traffic, we insert a Layer-2 RED [27] NFV element in the packet gateway to drop packets at the early stage of congestion. **Figure 8** compares the traffic pattern when with and without Layer-2 RED NFV element in a real experiment, where we set the bandwidth of bottleneck link to 200 Kbps. The result shows that with Layer-2 RED, the bandwidth utilization ratio could be increased to 100%.

6.3 eMBB: Content Delivery

Over the last decade, content delivery networks (CDNs) have played a significant role in hosting and distributing content to users. Thanks to its architecture that consists of multiple servers distributed geographically, content is replicated across the wide area, and hence is highly available. Several studies have demonstrated the effectiveness of CDNs in improving the quality-of-experience (QoE) by making applications and services faster and more reliable [28], [32]. This concept has helped many renowned companies to develop and to expand their revenue. CDNs can improve the access by caching and streaming content, with many distributed components collaborating to deliver content across different network nodes. CDN providers have general and distributed topologies around the world [46].

The idea of CDN as a service (CDNaaS) platform is the offer of a tool that allows different users to create their CDN slices, on top of different clouds, without writing a single line of code or deploying any server. **Figure 9** shows the main idea of CDNaaS platform. As depicted in this figure, the created CDN slice will

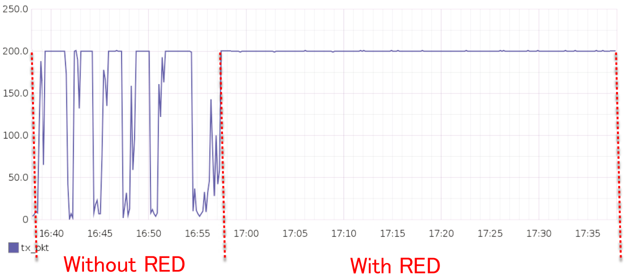

Fig. 8 Layer-2 RED control applied to mMTC traffic.

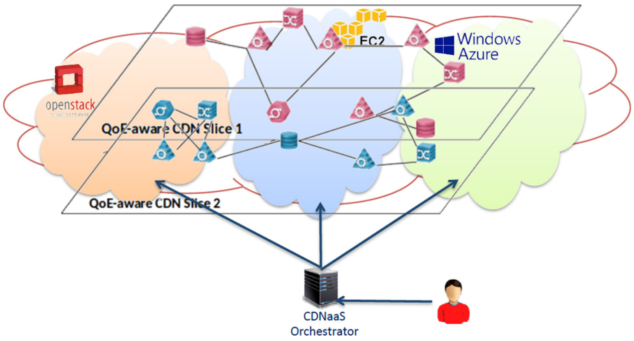

Fig. 9 CDNaaS platform high level diagram.

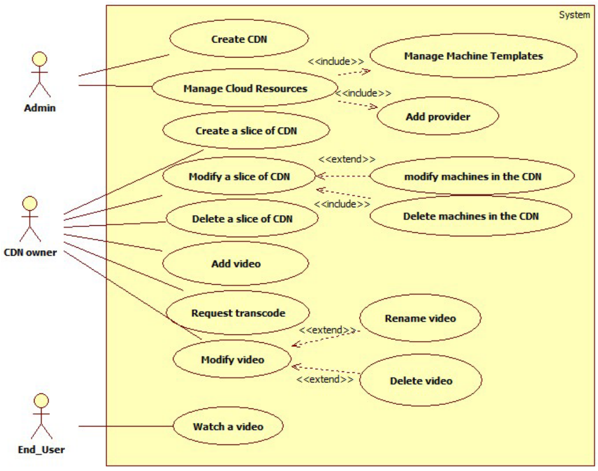

Fig. 10 Use-case diagram of CDNaaS platform.

run on different clouds and that for serving many users in the globe and by offering CDNs with high QoE. Using dedicated servers and their orchestrater, the users can create different CDN slices that are deployed in different clouds. This platform is designed to have the maximum level of flexibility for integrating with different public and private infrastructure as a service (IaaS) providers such as Amazon AWS service [21], Microsoft Azure [35], Rackspace [42] and OpenStack [40] in order to host different CDNaaS components.

Figure 10 shows the use-case diagram of CDNaaS platform. As depicted in this figure, we have mainly three kinds of users: i) admin user; ii) CDN owner user; iii) end user. An admin user is responsible for creating different CDN platforms, where different CDN slices can be created. Moreover, the admin user is also responsible for creating different components in the clouds, such as streamers, transcoders and caches, etc. The admin user is also responsible for including new cloud network providers to the system. The admin user uses the required credentials of each cloud network provider for including it to the CDNaaS platform. A CDN owner user is responsible for creating and managing different CDN slices in different cloud network. The CDN owner user is also responsible for managing and updating different videos in the system. The system offers user-friendly interfaces and enables the orchestration between different users and components. When a CDN slice is created, an end user can watch different videos in different locations through the web.

Figure 11 shows the sequence diagram of creating CDN slice. A CDN owner should authenticate at the orchestrator by offering the required credentials. The CDN owner specifies necessary CDN components in different network cloud providers. Then, an orchestrator automatically contacts the concerned providers for creating the different components of the CDN slice. As mentioned before, in each CDN slice, there is only one coordinator that is responsible for managing the whole CDN slices in different network providers. The information such as IP addresses, credentials is conveyed from different components to the coordinators. While private IP addresses may be used for the components located in the same cloud network with the coordinator, public IP addresses can be used for the ones located in the other networks.

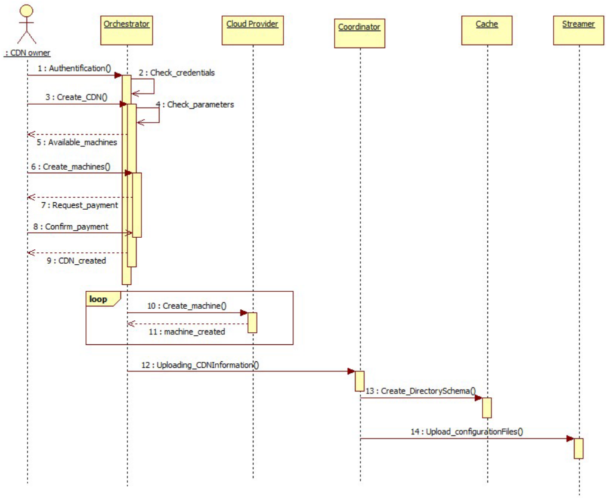

Fig. 11 Sequence diagram for creating and managing CDN slices.

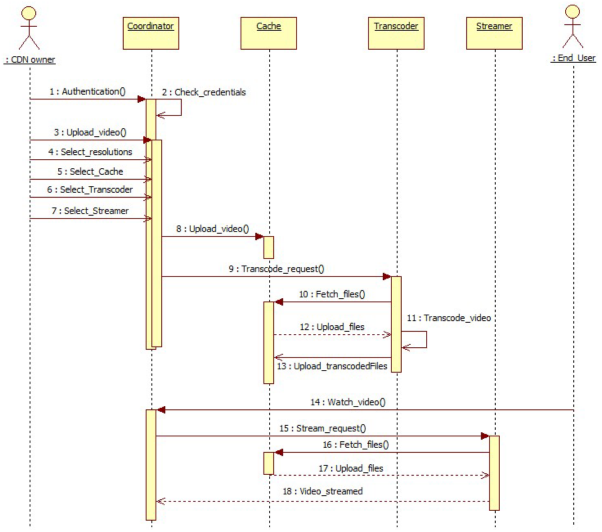

Fig. 12 Sequence diagram for creating and managing video contents.

Each CDN slice consists of one coordinator and a set of caches, transcoders, and streamer servers. The coordinator is responsible for the management of the CDN slice. It gives to the CDN slice owner the possibility to upload different video contents, as well as it ensures the communication among different CDN slice components within the same cloud or within different network clouds. **Figure 12** shows the sequence diagram of creating and managing videos. In this diagram, the CDN owner user is responsible for uploading and updating the videos, whereas the end user watches the videos by acceding first to the coordinator.

To upload new video or manage existing videos, the CDN user must be authenticated at the coordinator. For each video, the CDN owner user chooses the resolutions, and the cache, the streamer and the transcoder servers. The coordinator communicates with the cache server for caching the video and communicates with the transcoder server for transcoding the video on different resolutions. When an end user wants to watch a video, the coordinator communicates first with the streamer, which is responsible for streaming the content of videos from the caches.

7. Conclusion

In this paper, we introduce the emerging concept of network slicing that is considered one of the most significant technology challenges for 5G mobile networking infrastructure, summarize our preliminary research efforts to enable end-to-end network slicing for 5G mobile networking, and finally discuss the application use cases that should drive the designs of the infrastructure of network slicing.

We have originally defined network slicing as an isolated set of programmable resources to implement network functions and application services through software programs in early 2000s. We believe the two most significant features of network slicing are *resource isolation* and *programmability on resources* that are essential for accommodating individual software defined network functions and application services within each slice without interfering with the other functions and services on the coexisting slices. We have just begun extending the concept of network slicing to 5G mobile networking to accommodate applications with very different requirements, such as eMBB, mMTC and URLLC, to avoid interference among those, to guarantee the end-to-end quality of applications. We strongly believe we benefit from network slicing in realizing 5G mobile networking and beyond.

References

[1] 3GPP, available from ⟨http://www.3gpp.org/⟩.

[2] 5G Forum, available from ⟨http://www.5gforum.org/⟩.

[3] The 5G Infrastructure Public Private Partnership, available from ⟨https://5g-ppp.eu/⟩.

[4] Create Prototypes and Get to Market Faster Using Intel® Edison Technology, available from ⟨https://software.intel.com/en-us/iot/hardware/edison⟩.

[5] Fed4Fire, available from ⟨https://www.fed4fire.eu⟩.

[6] FLARE: Open Deeply Programmable Network Node Architecture, available from ⟨http://netseminar.stanford.edu/seminars/10_18_12.pdf⟩.

[7] IMT-2020 (5G) Promotion Group, available from ⟨www.imt-2020.cn/en/⟩.

[8] ITU, available from ⟨https://www.itu.int/⟩.

[9] LoRa Alliance, available from ⟨https://www.lora-alliance.org/⟩.

[10] Open vSwitch, available from ⟨http://openvswitch.org/⟩.

[11] sigfox, available from ⟨http://www.sigfox.com/⟩.

[12] GENI (2007), available from ⟨https://www.geni.net⟩.

[13] PlanetLab (2012), available from ⟨http://www.planet-lab.org⟩.

[14] PlanetLab (2012), available from ⟨https://www.planet-lab.eu⟩.

[15] Overview of 3GPP Release 8 V0.3.3 (2014-09), available from ⟨http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-08_description_20140924.zip⟩.

[16] FG IMT-2020: Report on Gap Analysis (2015), available from ⟨http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/T13-SG13-151130-TD-PLEN-0208%21%21MSW-E.docx⟩.

[17] Focus Group of IMT-2020 (2015), available from ⟨http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Pages/default.aspx⟩.

[18] Overview of 3GPP Release 12 V0.2.0 (2015-09), available from ⟨http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-12_description_20150909.zip⟩.

[19] Release 13 analytical view version Sept. 9th 2015 (2015), available from ⟨http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-13_description_20150917.zip⟩.

[20] USRP B210 (2015), available from ⟨https://www.ettus.com/product/details/UB210-KIT⟩.

[21] Amazon ec2 - virtual server hosting (2016), available from ⟨https://aws.amazon.com/ec2/⟩.

[22] Gebremariam, A.A., Chowdhury, M., Goldsmith, A. and Granelli, F.: Resource Pooling via Dynamic Spectrum-level Slicing across Heterogeneous Networks, *IEEE CCNC*, pp.1–6, Las Vegas, USA (2017).

[23] 4G Americas: The Voice of 5G and LTE for the Americas, available from ⟨http://www.4gamericas.org/⟩.

[24] Chen, B. and Morris, R.: Flexible control of parallelism in a multiprocessor PC router, *USENIX Annual Technical Conference* (2001).

[25] Du, P. and Nakao, A.: Application Specific Mobile Edge Computing through Network Softwarization, *IEEE International Conference on Cloud Networking* (*CloudNet*) (2016).

[26] Du, P., Putra, P., Yamamoto, S. and Nakao, A.: A context-aware IoT architecture through software-defined data plane, *IEEE Region 10 Symposium* (*TENSYMP*), pp.315–320 (2016).

[27] Floyd, S. and Jacobson, V.: Random early detection gateways for congestion avoidance, *IEEE/ACM Trans. Netw.*, Vol.1, No.4, pp.397–413 (1993).

[28] Gadde, S., Chase, J. and Rabinovich, M.: Web caching and content distribution: A view from the interior, *Computer Communications*, Vol.24, No.2, pp.222–231 (2001).

[29] Hansen, C.J.: WiGiG: Multi-gigabit wireless communications in the 60 GHz band, *IEEE Wireless Communications*, Vol.18, No.6, pp.6–7 (2011).

[30] Katsalis, K., Nikaein, N., Schiller, E., Favraud, R. and Braun, T.I.: 5G Architectural Design Patterns, *IEEE ICC Workshops: 3rd Workshop on 5G Architecture*, pp.1–6, Kuala Lumpur, Malaysia (2016).

[31] Kohler, E.: The Click modular router, PhD Thesis, MIT (2000).

[32] Krishnamurthy, B., Wills, C. and Zhang, Y.: On the use and performance of content distribution networks, *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement*, pp.169–182, ACM (2001).

[33] Merkel, D.: Docker: Lightweight linux containers for consistent development and deployment, *Linux Journal*, Vol.2014, No.239, p.2 (2014).

[34] METIS: The METIS 2020 Project - Laying the foundation of 5G, available from ⟨https://www.metis2020.com/⟩.

[35] Microsoft: Create linux and windows virtual machines in minutes (2016), available from ⟨https://azure.microsoft.com/en-us/services/virtual-machines/⟩.

[36] Nakao, A.: Network virtualization as foundation for enabling new network architectures and applications, *IEICE Trans. Communications*, Vol.93, No.3, pp.454–457 (2010).

[37] Nakao, A., Du, P. and Iwai, T.: Application Specific Slicing for MVNO through Software-Defined Data Plane Enhancing SDN, *IEICE Trans. Communications*, Vol.98, No.11, pp.2111–2120 (2015).

[38] NGMN: ngmn - next generation mobile networks, available from ⟨http://www.ngmn.org/⟩.

[39] Nikaein, N., Marina, M.K., Manickam, S., Dawson, A., Knopp, R. and Bonnet, C.: OpenAirInterface: A flexible platform for 5G research, *ACM SIGCOMM Computer Communication Review*, Vol.44, No.5, pp.33–38 (2014).

[40] Openstack: Open source software for creating private and public clouds (2016), available from ⟨https://www.openstack.org/⟩.

[41] 5GMF White Paper: 5G Mobile Communications Systems for 2020 and beyond (2016), available from ⟨http://5gmf.jp/wp/wp-content/uploads/2016/09/5GMF_WP101_All.pdf⟩.

[42] Rackspace: The industry leading open source technology (2016), available from ⟨https://www.rackspace.com/cloud⟩.

[43] Sasaki, K., Suzuki, N., Makido, S. and Nakao, A.: Vehicle Control System Coordinated between Cloud and Mobile Edge Computing, *IEEE SICE* (2016).

[44] Shenker, S., Zhang, L. and Clark, D.D.: Some observations on the dynamics of a congestion control algorithm, *ACM SIGCOMM Computer Communication Review*, Vol.20, No.5, pp.30–39 (1990).

[45] Taleb, T., Corici, M., Parada, C., Jamakovic, A., Ruffino, S., Karagiannis, G. and Magedanz, T.: EASE: EPC as a service to ease mobile core network deployment over cloud, *IEEE Network*, Vol.29, No.2, pp.78–88 (2015).

[46] Vakali, A. and Pallis, G.: Content delivery networks: Status and trends, *IEEE Internet Computing*, Vol.7, No.6, pp.68–74 (2003).

[47] Xiong, X., Zheng, K., Xu, R., Xiang, W. and Chatzimisios, P.: Low power wide area machine-to-machine networks: Key techniques and prototype, *IEEE Communications Magazine*, Vol.53, No.9, pp.64–71 (2015).

[48] Yamada, K., Kanada, Y., Amemiya, K., Nakao, A. and Saida, Y.: VNode infrastucture enhancement — Deeply programmable network virtualization, *2015 21st Asia-Pacific Conference on Communications* (*APCC*), pp.244–249, IEEE (2015).

Akihiro Nakao received B.S. (1991) in Physics, M.E. (1994) in Information Engineering from the University of Tokyo. He was at IBM Yamato Laboratory, Tokyo Research Laboratory, and IBM Texas Austin from 1994 till 2005. He received M.S. (2001) and Ph.D. (2005) in Computer Science from Princeton University. He has been teaching as an associate professor (2005–2014) and as a professor (2014–present) in Applied Computer Science, at Interfaculty Initiative in Information Studies, Graduate School of Interdisciplinary Information Studies, the University of Tokyo.

Ping Du received B.E. and M.E. degrees from University of Science and Technology of China in 2000 and 2003, respectively. He received his Ph.D. from the Graduate University for Advanced Studies in Japan in 2007. From 2008, he worked for the National Institute of Information and Communication Technologies (NICT) of Japan. Now, he works for the University of Tokyo as a project assistant professor. His research interests include optical network, network security, network virtualization etc.

Yoshiaki Kiriha is a project researcher in the University of Tokyo. He received his M.S. degree in Electrical Engineering from Waseda University in 1987. He then joined NEC, where he worked in the R&D division for over 20 years, and has been leaded in many projects related to distributed database systems, real-time systems, as well as Future Internet service & management systems. He has contributed continuously as a TPC member for almost of all IM & NOMS conferences from 2000. He has also served as a chair of TC on Information Communication Management IEICE 2010–2011.

Fabrizio Granelli is Associate Professor and Delegate for Education at the Department of Information Engineering and Computer Science (DISI) at the University of Trento (Italy). He is currently Director for Online Content of the IEEE Communications Society and was an IEEE ComSoc Distinguished Lecturer for the period 2012–2015. He received the Laurea (M.Sc.) and Ph.D. degrees from the University of Genoa, Italy, in 1997 and 2001, respectively. He was visiting professor at the State University of Campinas (Brasil) and at The University of Tokyo (Japan). He has authored or co-authored more than 170 papers on topics related to networking, with a focus on wireless communications and networks, cognitive radios and networks, green networking, and smart grid communications. He is the founder and general vice-chair of the First International Conference on Wireless Internet (WICON '05) and general chair of the 11th, 15th, and 18th IEEE Workshop on Computer-Aided Modeling, Analysis, and Design of Communication Links and Networks (CAMAD). He is TPC co-chair of the IEEE GLOBECOM Symposium on "Communications QoS, Reliability and Performance Modeling" in the years 2007, 2008, 2009, and 2012.

Anteneh Atumo Gebremariam is a Ph.D. Candidate at university of Trento, Italy. His current research focuses on abstraction, virtualization, and efficient resource utilization of future wireless networks, 5G Networks and beyond, applying the concepts of network function virtualization (NFV) and software-defined networking (SDN) paradigms. In addition, he is currently working on E2E resource slicing techniques in order to provide the proper environment for supporting various bandwidth hungry services/applications (e.g., IoT, M2M communication, video, etc.) on top of a single physical wireless infrastructure. His previous working experience in an industry, gives him a great perspective in applying his current research to the real world applications. He also collaborated with CREATE-NET Research Group, Trento, Italy on EU project called 5G Infrastructure Public Private Partnership (5G PPP). In addition, during his studies he spent a period abroad as a Visiting Student Researcher and as an intern in Stanford University and in Nokia Siemens Networks (NSN) respectively. Moreover, his collaboration work with different interdisciplinary research groups enabled him to develop his team working skills and also the ability to work independently.

Tarik Taleb received his B.E. degree in information engineering (with distinction), his M.Sc. and Ph.D. degrees in information sciences from GSIS, Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. He is an IEEE Communications Society (ComSoc) Distinguished Lecturer. He is a Member of the IEEE Communications Society Standardization Program Development Board. In an attempt to bridge the gap between academia and industry, he founded the IEEE Workshop on Telecommunications Standards: From Research to Standards, a successful event that was recognized with the Best Workshop Award by the IEEE Communication Society (ComSoC). Based on the success of this workshop, he has also founded and has been the Steering Committee Chair of the IEEE Conference on Standards for Communications and Networking. He is the General Chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference (WCNC '19) to be held in Marrakech, Morocco. He is/was on the Editorial Board of 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.

Miloud Bagaa received his M.E. and Ph.D. degrees from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2005, 2008, and 2014, respectively. From 2009 to 2015, he was a Researcher with the Research Center on Scientific and Technical Information (CERIST), Algiers, where he was a Member of the Wireless Sensor Networks Team, DTISI Division. From 2015 to 2016, he was granted a postdoctoral fellowship from the European Research Consortium for Informatics and Mathematics, and worked with the Norwegian University of Science and Technology, Trondheim, Norway. Currently, he is Senior Researcher with the Communications and Networking Department, Aalto University, Espoo, Finland. His research interests include wireless sensor network, Internet of Things, 5G wireless communication, security, and networking modeling.