Open-Source Multi-Access Edge Computing for 6G: Opportunities and Challenges

LIQIANG ZHAO[1], (Member, IEEE), **GUORONG ZHOU**[1], **GAN ZHENG**[2], (Fellow, IEEE),
CHIH-LIN I[3], (Fellow, IEEE), **XIAOHU YOU**[4], (Fellow, IEEE),
AND LAJOS HANZO[5], (Life Fellow, IEEE)

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[2]Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough LE11 3TU, U.K.
[3]Green Communication Research Center, China Mobile Research Institute, Beijing 100053, China
[4]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[5]School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Corresponding author: Lajos Hanzo (lh@ecs.soton.ac.uk)

ABSTRACT Multi-access edge computing (MEC) is capable of meeting the challenging requirements of next-generation networks, *e.g.*, 6G, as a benefit of providing computing and caching capabilities in the close proximity of the users. However, the traditional MEC architecture relies on specialized hardware and its bespoke software functions are closely integrated with the hardware, hence it is too rigid for supporting the rapidly evolving scenarios in the face of the demanding requirements of 6G. As a remedy, we conceive the compelling concept of open-source cellular networking and intrinsically amalgamate it with MEC, which is defined by open-source software running on general-purpose hardware platforms. Specifically, an open-source MEC (OS-MEC) scheme is presented relying on a pair of core principles: the decoupling of the MEC functions and resources from each other with the aid of network function virtualization (NFV); as well as the reconfiguration of the disaggregated MEC functions and resources into customized edge instances. This philosophy allows operators to adaptively customize their users' networks. Then, we develop improved networking functions for OS-MEC decoupling and discuss both its key components as well as the process of OS-MEC reconfiguration. The typical use cases of the proposed OS-MEC scheme are characterized with the aid of a small-scale test network. Finally, we discuss some of the potential open-source-related technical challenges when facing 6G.

INDEX TERMS Open-source cellular network, multi-access edge computing, 6G, network function virtualization, service-based architecture, management and orchestration.

I. INTRODUCTION

By integrating communication and information technologies, multi-access edge computing (MEC) [1]–[3] intrinsically amalgamates the computing and caching capabilities in the immediate proximity of the users, which reduces the transmission delay and improves the resource utilization at the edge. As a further benefit, the emergence of MEC alleviates both the potential data transmission congestion

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

of the transport network (TN) as well as the data processing burden imposed on the core network (CN). Therefore, in December 2014, the European telecommunications standards institute (ETSI) has initiated MEC standardization for continuously updating and improving the MEC reference architecture and its applications [4]. Many operators and vendors provide a convenient computing and caching environment at the edge by relying on ETSI's MEC architecture [5]–[7].

With the continual growth of the mobile Internet traffic, next-generation networks are expected to operate at rates

up to Tbps and delays lower than 1ms [8]–[10]. MEC is considered as a potential enabling technology of the sixth generation (6G), capable of satisfying enhanced service requirements [11]. For example, MEC can guarantee ultra-low transmission delay and satisfy flawless 4k/8k video broadcast by processing computation-intensive data or caching ultra-high-definition (UHD) videos at the edge [12], [13]. However, the exiting MEC reference architecture is built upon specialized hardware and its software functions are closely integrated with the hardware, which is inconvenient in the face of rapidly evolving scenarios, when aiming for satisfying the demanding requirements of 6G. As a remedy, the above challenges may be met by introducing the open-source (OS) cellular network concept into MEC for allowing operators to adaptively customize their users' networks.

Open-source software [14] has a very long history, which can be traced back to the organizations of SHARE and DECUS in 1950s. In the following years, ARPANET, UNIX, Linux and Android were developed for altruistic public sharing. By contrast, OS cellular networks are still in their infancy, despite the associated benefits. Explicitly, in the OS cellular network, an open infrastructure would enable flexible network reconfiguration based on the specific requirements of diverse scenarios, and open source software would enable users to develop the future network in a collaborative open-access manner. Such an OS cellular network has to rely on the integration of 6G candidate technologies, such as network function virtualization (NFV) [15], software-defined networking (SDN) [16], network slicing [17] and MEC [18], *etc*. Among them, SDN/NFV serve as the core drivers, which should additionally be integrated with MEC for cell-edge performance enhancement in support of customized radio access networks (RANs).

Substantial efforts have been invested in promoting the progress of OS networks [19]. The Open-Air-Interface (OAI) community founded in Europe exploited a fully-fledged fourth generation (4G) protocol stack based on commercial off-the-shelf (COTS) hardware. As a further advance, the Open5G community founded in China developed the concepts, technologies and platforms of open-source fifth generation (5G). The O-RAN alliance [20] founded by the global industry supported by academics has endeavoured to evolve RANs around the world. Additionally, the Linux Foundation (LF) launched the open network automation platform (ONAP) for orchestration. As for virtualization, the open platform of NFV (OPNFV) was conceived. Recently, there have also been a few open-source efforts on MEC [21]. For example, the Akraino project built by LF offers a pair of blueprints, namely the 5G MEC/slice and the micro-MEC system, but unfortunately it remains focused on the virtualization layer of the MEC reference architecture. Based on all these efforts, we present a new paradigm for developing an all-encompassing open-source MEC (OS-MEC) scheme.

The contributions of this paper are summarized as follows.

1) Using ETSI's MEC architecture as a springboard, this paper introduces an OS-MEC framework to facilitate customized services for any potential scenario in 6G network, which is built on a pair of key principles, namely on network decoupling and reconfiguration.

2) With reference to the service-based architecture (SBA) of the 5G core network (5GC) [22], a service-based MEC layer is developed for the OS-MEC scheme for decomposing the tightly coupled service functions into multiple independent network functions (NFs), thereby achieving MEC decoupling.

3) In order to integrate these decoupled NFs as required and to reconfigure a MEC system, the concept of templates and instances is implemented as part of the NF's management and orchestration (MANO) scheme, eventually generating a customized entity for supporting complete services.

4) Several typical use cases are demonstrated based on our test network for validating its flexibility and customization. Our novel contributions are also contrasted at a glance boldly and explicitly to the state-of-the-art in Table 1.

The rest of this paper is organized as follows. We first introduce the basic concepts and key technologies of OS cellular networks in Section II. In Section III, we present our OS-MEC framework, its service-based MEC layer, interface, and the NF's MANO. In Section IV, our small-scale test network, our use cases and our demonstrations are presented. In Section V, we elaborate on some potential technical challenges. Finally, we conclude in Section VI.

II. THE OPEN-SOURCE NETWORK

In this section, our vision of an OS cellular network relies on four key elements: the cloud domain including CN; the edge domain including distributed MEC servers; the heterogeneous terminal domain; and the network domain connecting the cloud-edge-terminal domain, which includes a RAN linking the edge and terminal domains, and a transport network linking the cloud and edge domains, as shown in Fig. 1.

In the cloud domain, the OS CN architecture has been well documented. For example, SBA has been identified as a basic element of the 5GC architecture in the 3rd Generation Partnership Project's (3GPP) standards, which exhibits a high grade of flexibility and scalability. In SBA, the originally coupled network functions are partitioned into multiple independent functional blocks, where each of them provides one or more services that can be used by any of the other functional blocks. Therefore, all functional blocks can communicate with each other in order to support the users with customized applications, as if they were connected to a service bus. In the edge domain, the OS-MEC concept will be discussed in the next section. However, in the network domain, the OS RAN architecture has not as yet been defined, because it is hampered by its increased real-time complexity requirement. To achieve this, some promising solutions, such as RAN decoupling and reconfiguration, have been put forward. Firstly, the tightly coupled RAN elements, functions

TABLE 1. A comparative summary of contributions of the salient existing works.

| Feature	OAI [24]	Open5G [25]	O-RAN [20], [26]	ONAP [27]	OPNFV [28]	ETSI's MEC [4]	Akraino [21]	SDEC [29]	NDN-ECC [30]	Proposed
Focusing on MEC						✓	✓	✓	✓	✓
Applied in 6G network			✓	✓	✓		✓		✓	✓
Open-source architecture	✓	✓	✓	✓	✓		✓	✓	✓	✓
Application and resource decoupling using NFV	✓	✓		✓	✓	✓	✓	✓		✓
Network functions decoupling by SBA				✓						✓
Using MANO technology		✓	✓	✓		✓	✓	✓		✓
Reconfiguration by templates and instances										✓
Supporting multiple access on RAN side		✓				✓		✓		✓
Focusing on upper layers of architecture			✓	✓						✓
Using Docker and Kubernetes technologies				✓			✓			✓
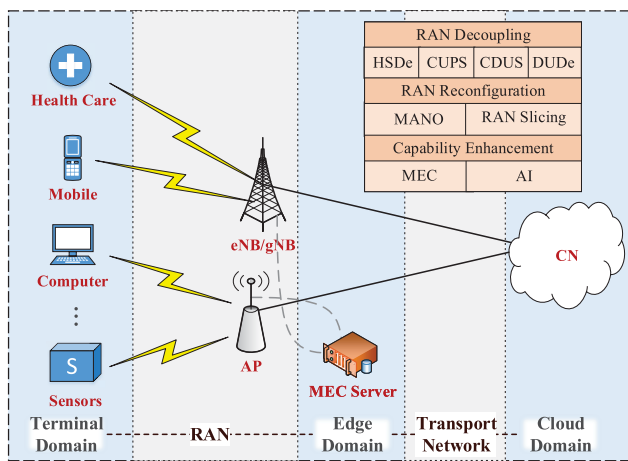
FIGURE 1. OS cellular network.

and resources of conventional networks have to be completely split. Then, they could be reassembled dynamically according to a user's requirements, thus providing customized virtual RANs. In a nutshell, the OS concept facilitates a convenient LEGO-block based construction of 6G RANs.

A. RAN DECOUPLING

The tightly coupled 4G cellular networks rely on a rigid closed operating system. Hence at least four different RAN decoupling methods have been conceived for orchestrating a beneficial degree of openness in 6G, namely hardware/software decoupling (HSDe) [31], control plane and user plane splitting (CUPS) [32], central unit and distributed unit splitting (CDUS) [33], as well as downlink and uplink decoupling (DUDe) [34]. Since they have different pros and cons, there is no wide agreement concerning the best OS RAN architecture. It is worth noting that HSDe and CUPS constitute the basis of OS cellular networks, while CDUS and DUDe further exploit the flexibility and openness of cellular networks.

HSDe decouples the resources and functions from the physical facilities. Given the mature family of NFV technologies, the network resources can be abstracted and shared among different users, and diverse services can be supported by the same infrastructure, thus separating the functionalities from the underlying infrastructure.

The idea of CUPS stems from SDN, which decouples the control signaling from the data transmission. Then, it further expands to cellular networks by deploying control base stations (CBS) in the most friendly lower frequency band and data base stations (DBS) in higher frequency bands to form the control plane (CP) and user plane (UP), respectively.

CDUS represents one of the key technologies for the next-generation cloud RAN, which focuses on separating the traditional evolved node base station (eNB) into the radio transceiver (distributed unit) and the logical center (central unit) from the perspective of the OS protocol stack.

The tightly coupled downlink (DL) and uplink (UL) transmission limits the flexibility of a user's connections. Upon adopting DUDe via dual-connectivity, where a user connects to a macro base station (MBS) and a small base station (SBS) for its DL and UL delivery respectively, flexible user association and high energy efficiency can be achieved.

B. RAN RECONFIGURATION

Upon using RAN decoupling technologies, conventional RANs have been softwarized and split into independent virtual NFs. Then, to support diverse application (APPs), only the necessary decoupled NFs will be chosen and assembled to construct RANs on a demand basis. For reconfiguring 6G RAN efficiently, there have been at least two possible solutions, namely RAN slicing and MANO.

RAN slicing [17] allows the virtual NFs and radio resources in RAN to be dynamically allocated to various services. Explicitly, when a specific service request arrives, RAN slicing selects the necessary NFs using the resources allocated, and then encapsulates them to form a customized virtual network.

MANO is also an effective technique of achieving RAN configuration, and it can be applied more flexibly to different network scopes than RAN slicing. Among them, the management part is responsible for the overall monitoring of RANs, including life cycle management, fault detection, accounting and security monitoring [35]. The orchestration part [36] supports the NFs and uses the underlying resources for reassembling the RAN, which mainly includes two sub-parts: choosing the appropriate NFs and appropriately scheduling them for a certain service.

C. RAN CAPABILITY ENHANCEMENT

MEC supports local computing and storage resources, which can be integrated with the open RAN both for providing the users with low-latency services and for reducing the transport network's burden [37]. Although the OS RAN architecture has not as yet been defined, an OS-MEC scheme will be established and detailed in our paper, which relies on the above decoupling and reconfiguration technologies.

Furthermore, 6G will have an ever increasing complexity in support of its compelling applications, heterogeneous networking architecture, and diverse resources. Hence the conventional network management methods (*i.e.*, deployment, resource allocation and operation) associated with human-controlled philosophies will become inadequate, calling for sophisticated artificial intelligence (AI) techniques [9] in the RAN, for achieving self-organizing, automated operation and CapEx/OpEx savings.

III. THE OPEN-SOURCE MEC

In this section, we incorporate the fundamental concepts of OS networks into MEC by relying on decoupling and reconfiguration, and propose the OS-MEC concept for improving the openness and flexibility of the emerging MEC systems. Explicitly, by developing a service-based MEC layer, we decompose the tightly coupled service functions into multiple independent NFs and support them flexibly using a simple service-based interface (SBI). Next, by introducing the convenient template and instance concept, NF's MANO is proposed for reconfiguring MEC, in which the OS-MEC's templates are predefined and then instantiated when needed.

A. OS-MEC FRAMEWORK

The OS-MEC framework conceived is shown in Fig. 2. The entire framework includes the application plane and MANO plane, where the former is mainly responsible for data processing and transmission, involving the infrastructure layer, virtualization layer, service-based MEC layer, and application layer from bottom to top. By contrast, the latter is composed of the virtualized infrastructure manager (VIM) and MANO, which completes the coordination and resource management of the OS-MEC system. The pair of adjacent layers communicate with each other through standard interfaces and cooperate for completing the MEC services requested by users.
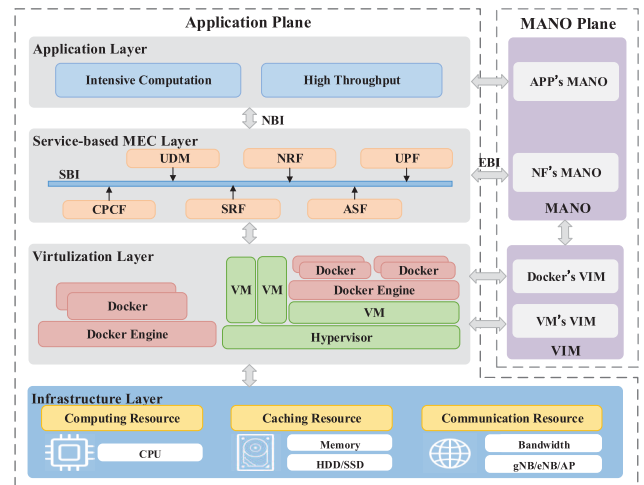

FIGURE 2. OS-MEC framework.

As seen in Fig. 2, the infrastructure layer is at the lowest level, covering all computing, caching and communication resources of the system. Specifically, the central processing unit (CPU) provides high-performance computing power for the service-based MEC layer right below the application layer at the top of Fig. 2. The memory and the hard disk drive (HDD)/solid state drive (SSD) are the main components of the caching resources in the infrastructure layer of Fig. 2. The communications resources of this layer include the bandwidth as well as the eNB, next-generation node base station (gNB) and access point (AP), as shown in the bottom layer of Fig. 2.

At the virtualization layer of Fig. 2, based on the NFV concept, the underlying three-dimensional resources may be decoupled from the dedicated hardware and abstracted into a resource pool that could be shared by various NFs at the service-based MEC layer of Fig. 2. Then, OS-MEC creates multiple Docker containers or virtual machines (VMs), which are run on the underlying resource pool and could simultaneously support different NFs for providing customized services [38], as seen at the virtualization layer of Fig. 2.

The service-based MEC layer of Fig. 2 constitutes the core part of OS-MEC, which includes a unified SBI and diverse NFs, where the SBI can connect these NFs together based on the unified stateless hypertext transfer protocol (HTTP) for ensuring that they can communicate directly with each other when needed. Furthermore, we decouple the originally centralized service functions into independent NFs. Specifically, the user plane function (UPF) of the service-based MEC layer in Fig. 2 is borrowed from 5GC for OS-MEC, in order to lend the 5G new radio (NR) capabilities to OS-MEC. Subsequently, to enhance OS-MEC, we download several other NFs at the control plane of 5GC and develop several new NFs. These NFs are transparent to each other and can also be combined at will, so that they could be activated and released by all users instantly in support of their customized services.
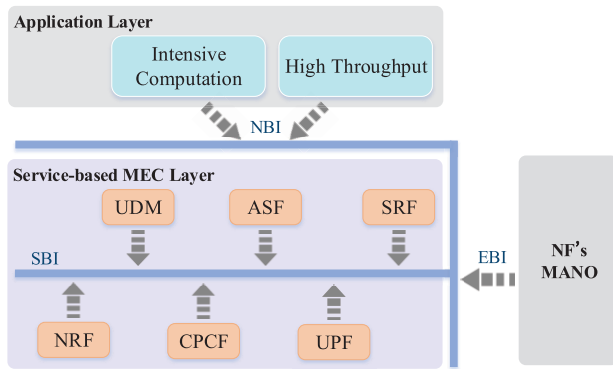
FIGURE 3. A detailed view of the service-based MEC layer of Fig. 2.

Actually, the user does not have to rely on the specific details of OS-MEC. Instead, they will use various APPs at the application layer to gain access to customized services. For simplicity, only two APPs are considered in this paper for processing caching and computing requests, respectively, *i.e.*, the high throughput and the intensive computation scenarios of the application layer seen at the top of Fig. 2.

Additionally, the MANO plane at the right of Fig. 2 is responsible for managing these NFs as well as APPs, and for scheduling the VIM to allocate resources. As mentioned earlier, MANO is in charge of MEC reconfiguration, where the OS-MEC's templates and instances are proposed for implementing flexible reconfiguration, as and when required. The VIM of the MANO plane in Fig. 2 manages the virtualized resources according to MANO's commands, so as to ensure that appropriate computing, caching and communication resources are supplied for the upper layers of Fig. 2.

Since there is rich literature on the design and implementation of both the virtualization layer and on the infrastructure layer [4], [21], [24], [25], [28]–[30], we mainly consider the service-based MEC layer and the MANO plane of OS-MEC.

B. MEC DECOUPLING

In order to decouple the MEC functions, we conceive the service-based MEC layer of the OS-MEC framework, which evolved from the traditional MEC layer, mainly composed of the SBI and the service-oriented NFs. We extract the service-based MEC layer from Fig. 2, and portray its detailed structure in Fig. 3.

The SBI connects all the NFs together for facilitating their interface programming and implementation. The northbound interface (NBI) is an application program interface (API) provided for the application layer to interact with the service-based MEC Layer in order to obtain services. The east-bound interface (EBI) is an API for NF's MANO to interact with the service-based MEC Layer, which is mainly used to manage and orchestrate both the resource allocation and life cycle of NFs. On the basis of using the same HTTP, we design a unified Representational State Transfer-ful (RESTful) [23] API for the SBI, NBI and EBI,

which is lightweight and can be read both by people and machines easily. Thus, the service-based MEC layer provides direct communications between the NFs, the MANO and the NFs, as well as the NFs and APPs, thereby reducing the interface-complexity and the protocol-inconsistency in the traditional MEC architecture [18].

In this paper, all service-oriented NFs are independently deployed in the Docker containers mentioned in Fig. 2 in order to become virtualized. There are six basic NFs in our proposed OS-MEC of Fig. 3. Specifically, the MEC NFs learned from 5GC include the unified data management (UDM), the NF repository function (NRF) and UPF seen in Fig. 3. Furthermore, the communication protocol conversion function (CPCF), the service registry function (SRF) and the application selection function (ASF) of Fig. 3 are newly developed after considering the context of OS-MEC. Below we briefly describe the functional differences of UDM and NRF applied both in OS-MEC as well as 5GC, followed by the portrayal of our proposed CPCF, SRF and ASF of Fig. 3.

- UDM is used for unified data storage and management. In contrast to the 5GC UDM, the OS-MEC UDM does not have to deal with the issue of data structure compatibility because of its semi-homogeneous data. It mainly manages the data through the MySQL database management system [39], and implements data insertion, search, update and deletion in the data tables of specific services. OS-MEC UDM can also provide standardized interfaces for SRF and ASF to ensure flexible access and real-time data update.

- NRF is a repository provided for centralized storing of NFs. OS-MEC further decouples the 5GC NRF, thus the OS-MEC NRF is only responsible for the storage of NFs and APPs, while the service registration is repackaged as SRF, and the service discovery is integrated into MANO. Due to the limited edge resources, general NFs (including SRF, CPCF, UDM and NRF) use local storage, while dedicated NFs, *e.g.*, ASF and APPs are stored in the remote image repository. As for the latter, the OS-MEC server will rely on specific NF and APP images to provide services only when needed.

- CPCF can complete the communication protocol conversion. In the OS-MEC system, the stateless HTTP is mainly used for data exchange between NFs. However, the protocols of users requesting edge services are heterogeneous, so CPCF has to analyze and re-encapsulate them into a unified HTTP protocol for supporting compatibility of the system.

- SRF is responsible for the service registration of new APPs. When a new APP gets to OS-MEC, SRF first registers the APP to UDM and updates the APP-related parameters to data tables for improving the information exchange between APP and NFs. Then, SRF stores the APP in the remote NRF image repository and provides an external access interface for allowing all the other edge servers to use the image.
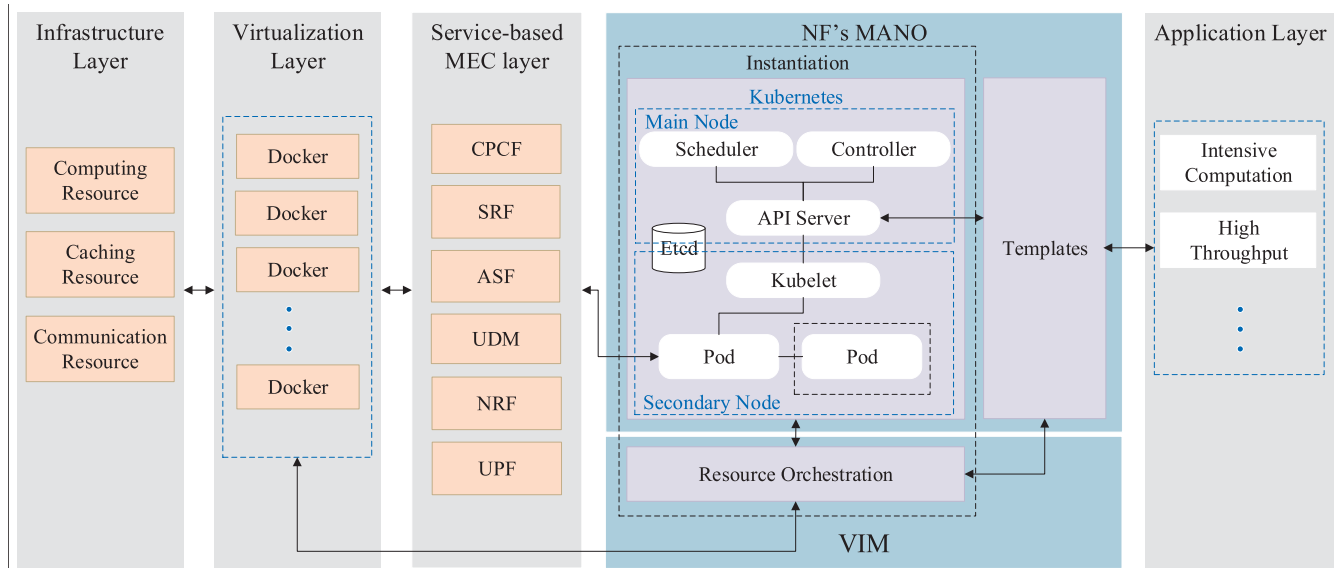
FIGURE 4. A detailed view of the NF's MANO of Fig. 2.

- ASF directly interacts with APPs through NBI for service selection. Specifically, compelling services could be provided in the form of APPs, such as intensive computation and high throughput services. Acting as the middleware between MEC APPs and other NFs, ASF is used both for the specific location of the requesting APPs and for communication between APPs and the other basic NFs.

The NF's MANO of Fig. 3 can flexibly schedule the NFs according to the specific service types, since the unified HTTP is used as EBI between the service-based MEC layer and NF's MANO, and is also used as SBI in the service-based MEC layer. With all the NFs deployed in the Docker containers in our solution, Kubernetes [40], the open-source implementation of the container cluster management, is advocated for centrally managing all the Docker containers playing the role of MANO. Its specific architecture will be introduced in the next subsection.

Based on the above NFs, we are ready for characterizing the basic functionalities of OS-MEC. Hence anyone could develop his/her own NFs for supporting new applications.

C. MEC RECONFIGURATION

As the MEC management is not directly related to the reconfiguration of OS-MEC, the joint management of MEC and NFV proposed by ETSI can still be used in OS-MEC, albeit the conception of service-based MEC layers is bound to impose challenges on the orchestration scheme. Therefore, by means of Kubernetes, we present OS-MEC's templates and instances of NF's MANO, which supports the flexible scheduling of NFs, and any further MEC reconfiguration. More explicitly, some templates are pre-defined for the corresponding applications, as exemplified by a computation-intensive template. However, when no user asks for this

application, we do not allocate the related resources to the template, hence, the template is empty. By contrast, as shown in Fig. 4, if a user requests this application, we shall activate the template by assigning computing, caching and communication resources as an instantiation.

1) NF's MANO

A detailed view of the NF's MANO extracted from Fig. 2 is shown in the middle (blue) part of Fig. 4, where the right side represents the predefinition and selection of templates, while the left side represents the templates' instantiation based upon Kubernetes [40], which is a container-based cluster management platform. The VIM seen at the bottom right of Fig. 4 is used for scheduling the underlying resource pool to arrange for the immediate resource deployment for instantiation, so as to avoid the assignment of excessive resources.

Observe in Fig. 4 that there is always a main node and either one or several secondary nodes in Kubernetes, where the former is in charge of managing Kubernetes, while the latter is in charge of the orchestration of NFs. In a secondary node, there are two basic components dedicated to instantiation in NF's MANO, namely Pod and Kubelet as seen in Fig. 4. As defined in [40], Pod is the smallest unit of deploying and monitoring NFs in Kubernetes, which can run several Docker containers simultaneously. Kubelet [40] of Fig. 4 maintains the life cycle of Pods and the communication between the master and the secondary nodes. In the master node, there are three basic components, namely the scheduler, controller and API server. The API server provides the only entry for secondary nodes to the application layer, *i.e.*, Pods communicate with the given templates selected by the user through the API server for implementing MEC reconfiguration. The Scheduler and Controller of Fig. 4 manage and monitor all secondary nodes, *e.g.*, for fault detection. Moreover, there is
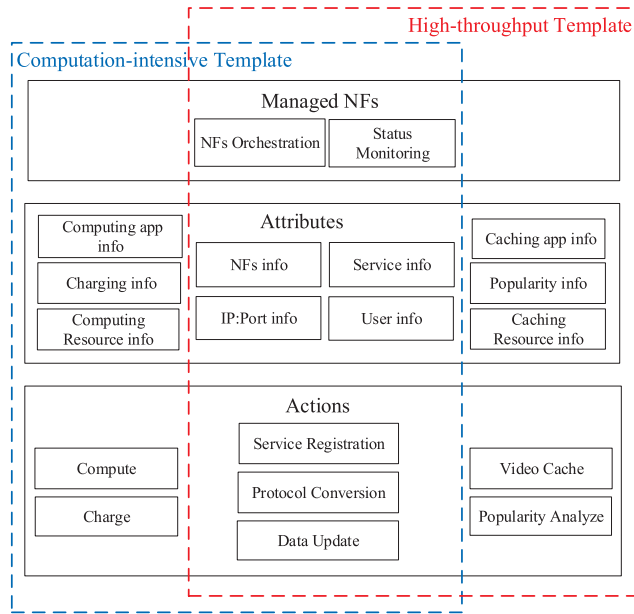
FIGURE 5. A detailed view of the OS-MEC templates introduced at the right of Fig. 4.

a Kubernetes' data center represented by Etcd of Fig. 4 for storing the states of the above components.

2) OS-MEC TEMPLATE

The template seen close to the right of Fig. 4 is provided for extracting the common characteristics of a class of problems. Given this idea, a static OS-MEC template is predefined, *e.g.*, for selecting appropriate NFs and for predefining the related parameters according to the specific type of APPs. Again, we consider two APPs, namely high throughput and intensive computation, as seen at the right of Fig. 4. These are examples of describing the OS-MEC templates, where the former is provided for high-throughput multimedia video services, while the latter is mainly for computing intensive services. Naturally, there are some differences between the two templates in terms of the NFs selection and data table definition in UDM, as detailed in Fig. 5.

Explicitly, observe in Fig. 5 that the OS-MEC template is composed of three tiers, including Managed NFs, Attributes and Actions. In this context, all NFs have to communicate directly with the Managed NFs at the top layer, which are responsible for monitoring the templates' status and orchestrating NFs. The middle-layer of Fig. 5 representing the Attributes acts as the database of templates, where the associated parameters are stored in the UDM's data tables. Finally, the underlying Actions seen at the bottom of Fig. 5 are provided for completing a range of customized functions by selecting appropriate NFs as well as APPs and for updating the predefined UDM parameters in real time.

We partition the computation-intensive and the high-throughput templates into the shared parts (overlap of the templates in Fig. 5) and the dedicated non-overlapping parts

of each. The shared part is irrelevant for the APPs, it essentially ensures the normal operation of the system, while the dedicated template is the key for supporting OS-MEC in providing specific services for the corresponding APP. For example, both the computation-intensive and the high-throughput templates have to complete service registration provided by SRF, protocol conversion completed by CPCF and data updates by UDM. However, the dedicated ASF and APP of the computation-intensive template are responsible for completing the computations based on the source-data, as seen at the bottom left of Fig. 5 and for setting up a charging function. By contrast, the dedicated ASF and APP of the high-throughput template process video caching and analyze the video popularity, as observed at the bottom right of Fig. 5.

3) OS-MEC INSTANCE

The OS-MEC template of Fig. 5 cannot directly provide services and we do not allocate the related resources for the template. As a result, users can only gain access to the edge services through instances. The instantiation determines whether the OS-MEC system can run stably, which is the key for reconfiguring MEC.

The instantiation process mainly includes the following three steps. Firstly, when a user requests edge services, such as intensive computation and high throughput, the MANO of Fig. 4 configures the operating environment of NFs, *i.e.* the API Server of Fig. 4 locates the idle Pod and updates the relevant information to Etcd of Fig. 4, which provides basic environmental support for instantiation. Secondly, the MANO of Fig. 4 schedules VIM for allocating the specified virtual computing, caching and communication resources to users. Finally, under the specifically configured operating environment and the resources allocated, MANO actually runs the predefined NFs and APPs of the template, *i.e.*, Kubelet runs the Pod that specifically deploys NFs and APPs. Therefore, the dedicated MEC instance has been created.

In Fig. 6, we summarize the workflow of OS-MEC, commencing from the instant when a user sends a service request until the edge service is instantiated. Specifically,

1) when a user sends a service request to the OS-MEC, such as the intensive computation service request, CPCF firstly actions the protocol identification.
2) Upon receiving a HTTP request, CPCF does not need to perform the protocol conversion; otherwise, it re-encapsulates the service request using HTTP. In this way, CPCF can send the unified HTTP request to MANO.
3) Then, MANO selects the computation-intensive template for the user. Explicitly,
   a) the predefined parameters of this template are added to the specific data table in UDM.
   b) By querying the NFs and APP information specified by the template in UDM, MANO selects the appropriate NFs and APPs stored in NRF.
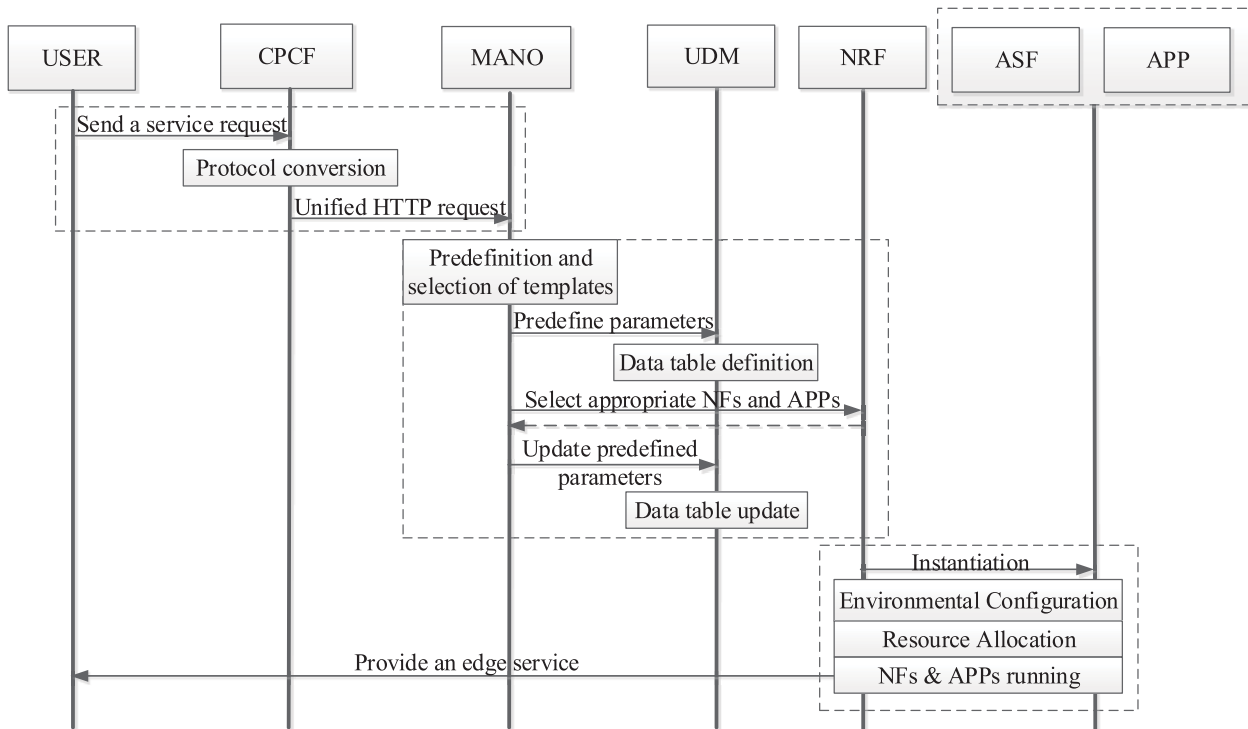
FIGURE 6. OS-MEC workflow.

c) MANO updates the predefined parameters in UDM's data table, completing the selection of templates.

4) Nextly, the instantiation process is implemented by the following three steps: the environmental configuration, resource allocation, and NF activation and APP activation.

5) Finally, OS-MEC provides a computation-intensive edge service to the user in the form of APP by the service interface.

At this point, we have succeeded in creating the OS-MEC scheme by completing the MEC decoupling and reconfiguration.

IV. DEMONSTRATION IN TEST NETWORK

In this section, we critically appraise the proposed OS-MEC scheme in the test network deployed at Xidian University and then characterize a pair of use cases, namely the intensive computation and the high throughput scenarios. Our test network is shown in Fig. 7. A 3GPP R10 based cellular network including an evolved packet core (EPC) and six eNBs was built based on open-source software provided by OAI [41] and general-purpose hardware, namely an Intel Core i7-7700@3.6GHz CPU and 16 GB random access memory (RAM). A commercial wireless router was used as an AP to deploy a WiFi network, which can support multiple access. Then, we deploy all the network elements in Docker containers to arrange for them to become virtualized, yielding a virtualized EPC (vEPC) and virtual NFs (vNFs). Subsequently, the user plane of both the serving gateway (SGW)

and of the packet data network gateway (PGW) is moved from vEPC into the OS-MEC server using an Intel Core i7-7567U@3.5GHz CPU and 32 GB RAM. The control plane, *i.e.* the home subscriber server (HSS) and the mobility management entity (MME), remains in the vEPC. Finally, based on Kubernetes, NF's MANO including its templates and instances is implemented to furnish users with customized services.

For instance, in the high throughput use case providing a video caching service, the user firstly accesses the test network of Fig. 7 by selecting an appropriate access mode, *e.g.*, eNB. Then, the network has to confirm, whether the requested video is stored in the OS-MEC server. If so, the high throughput service may indeed be directly completed in the OS-MEC server by the specific workflow seen in Fig. 6. By contrast, if the requested video cannot be found in the OS-MEC server, then the vSGW and vPGW in the OS-MEC server will connect to the cloud server of the campus network through the switch to access the required resources.

In the intensive computation use case, we provide three services, *i.e.* sum and prime sum computation as well as face recognition. These three services are requested in three independent instances respectively, which are created by the same computation-intensive template though. Therefore, three containers have to be started for the above instances, respectively. By contrast, in the high throughput use case providing a video caching service, a user watches his/her video cached at the OS-MEC server, which only creates a single instance and runs a single container.
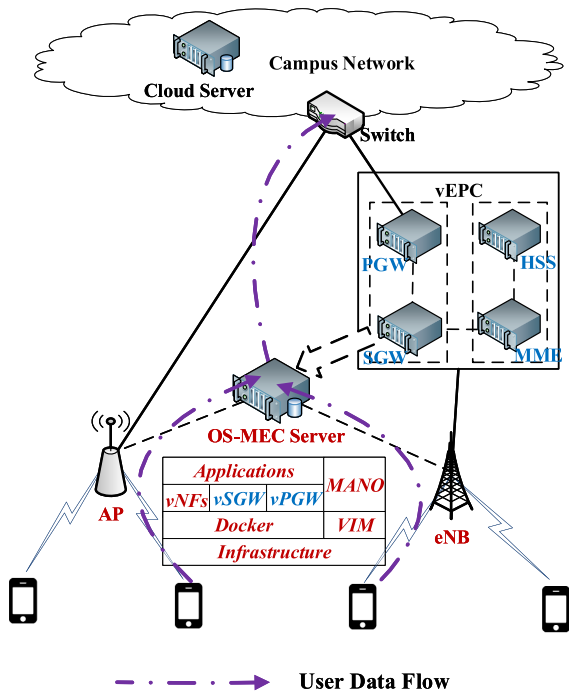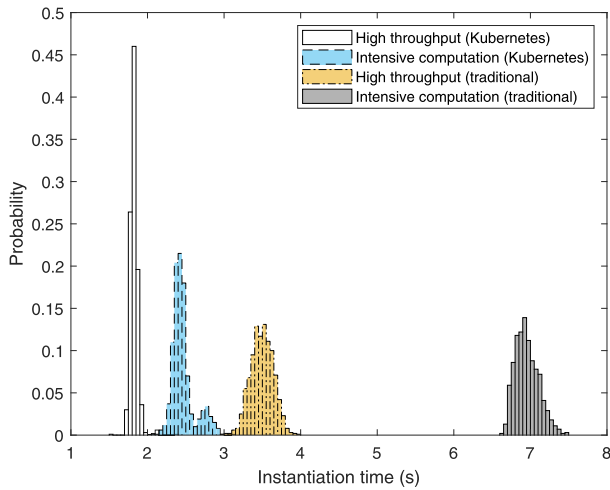
FIGURE 7. Test network.

FIGURE 8. Instantiation time-duration for computation-intensive and high-throughput services.

FIGURE 9. Computing time of three computation-intensive services.

Fig. 8 shows a histogram for 1000 experiments for characterizing the resultant instantiation time-duration both in the proposed Kubernetes method and in the traditional script running method used in [41], where we simultaneously support three computation-intensive services to test the instantiation performance. For both applications, the average instantiation duration of Kubernetes is always lower than that of the traditional script, because the Kubernetes method can run multiple containers in parallel, while the traditional method only runs them separately through scripts. Moreover, the intensive computation APP takes a little longer to instantiate, because
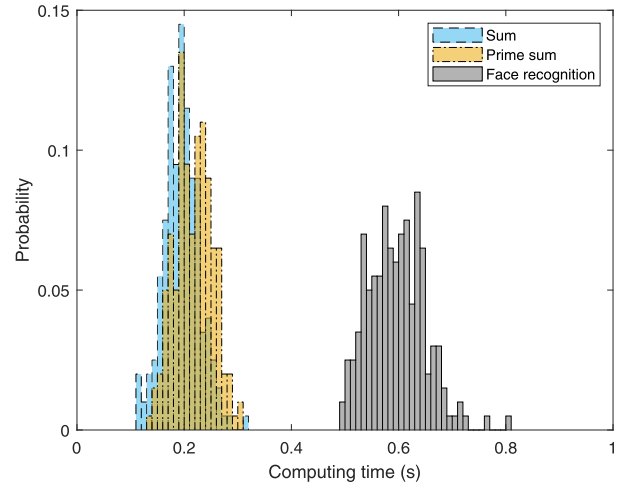
it starts three containers at the same time, while the high throughput APP only runs a single container. However, when using our Kubernetes method, their difference in time duration is small. This allows the operators to easily incorporate new NFs or services as and when needed without degrading the user experience.

To provide further insights, Fig. 9 portrays the comparison of the computing time of three computation-intensive services by a histogram for 1000 experiments. Since each picture has to be transmitted from mobile phones to the OS-MEC server and processed by the file transfer protocol (FTP), the average computing time of face recognition is higher than that of the other two services.

Fig. 10 analyzes the CPU and memory usage by calculating the prime sum and face recognition in turn. Since much more data have to be processed during face recognition, the CPU load of face recognition is about four times higher than that of prime sum computations. The face recognition also has to store more data than the prime sum computation, so it requires more memory in Fig. 10(b) (83.9MB). Then, whenever the network completes data processing, the CPU is freed up immediately and the computing resources are released. By contrast, the memory will not be freed up unless we handle it manually. For example, as seen in Fig. 10(a), the CPU is released immediately once the prime sum service is completed at 9:20:28. However, in Fig. 10(b), the memory remained reserved, until we released it manually at 9:20:30. Therefore, the limited edge computing and caching resources of OS-MEC are used flexibly and efficiently.

Fig. 11 quantifies the transmission and computing time of the high-throughput service, in which different-length videos are cached at the edge or cloud. Upon increasing the video-length, we can see that the gap of the transmission time between the cloud and the OS-MEC scheme gradually increases, which shows that the advantage of our proposed scheme is gradually enhanced. Since the computation loads of the high-throughput service are limited, the computing
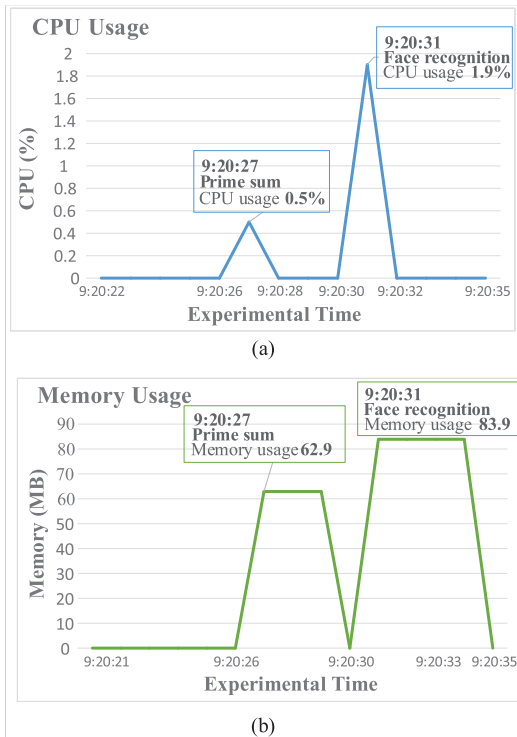
FIGURE 10. (a) CPU usage for intensive computation; (b) memory usage for intensive computation.
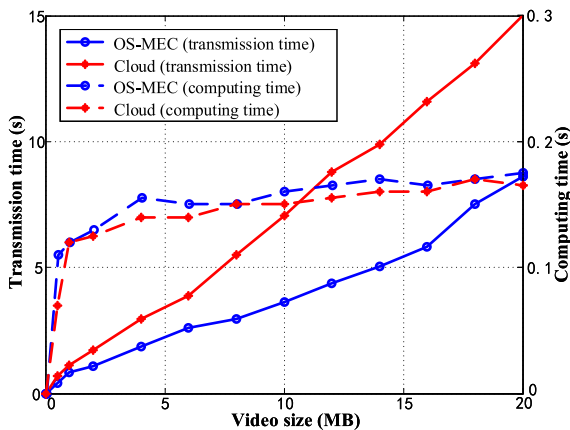

FIGURE 11. Transmission and computing time for high throughput.

time of both the cloud and of the edge is only about 0.14s on average, and their difference is small. Furthermore, the increase of video size has little effect on the computing time, which is basically in a stable state.

V. TECHNICAL CHALLENGES

As mentioned above, the rapidly evolving scenarios and requirements of the 6G network provides a compelling opportunity for introducing the OS cellular network concept into MEC, which allows the operators to adaptively customize the users' networks at the edge based on the proposed OS-MEC philosophy. Our solutions have the potential of

influencing the standardization of the MEC architecture for next-generation networks. However, there are still some open technical challenges in the research of 6G-oriented OS cellular networks having a MEC functionality, which need further research and global discussion.

A. CLOUD-NETWORK-EDGE-TERMINAL COLLABORATIVE OS CELLULAR NETWORK

In recent years, substantial progress has been made in the research of OS networks. For example, due to the proposal of the 5GC SBA, the OS CN architecture has been studied quite widely; the proposed OS-MEC scheme provides a complete overview of the edge-based OS architecture and its implementation. Nevertheless, the OS RAN architecture has not been clearly defined due to its complex air interfaces, heterogeneous cell structure and incompatible network protocols. Moreover, an integrated OS cellular network paradigm involving collaborations between the cloud, the network, the edge and the terminal is inevitable in the 6G network of the near future. Whether the existing OS network components can be integrated and work together still needs further testing.

B. EFFECTIVE INTEGRATION WITH 6G SCENARIOS

The 6G system will further improve the existing 5G networking scenarios [12], [13], with the objective of further improving ultra-reliable low latency communications (URLLC). It will also embrace new paradigm shifts, *e.g.* space-air-ground integrated networks [42], terahertz (THz) communication [43], *etc*. This requires that the future 6G OS network must be sufficiently flexible and intelligent for seamlessly supporting new scenarios and components, or for upgrading the existing functions without causing global disruption to the architecture. However, the proposed OS-MEC had to be developed and operated on the existing general-purpose hardware and open-source software platforms. Therefore, its integration with the future 6G application scenarios and paradigms needs further research.

C. OPEN-SOURCE MARKET FRAGMENTATION

The research of OS network platforms has attracted the attention of numerous organizations and institutions. Numerous OS network platforms have been developed, such as O-RAN, ONAP, OPNFV and the proposed OS-MEC, *etc*. However, the availability of a bewildering gamut of platforms can lead to the fragmentation of the open-source market, hence resulting in interoperability problems. In order to circumvent these problems, on the one hand, a unified open-source standard should be established for supporting openness and interoperability. On the other hand, it requires closer cooperation among open-source organizations to coordinate all the OS network platforms in the near future.

D. CYBERSECURITY AND INTELLECTUAL PROPERTY PROTECTION

The 6G-oriented OS cellular network architecture is expected to integrate all operating networks and provide users with

customized capabilities including mobile services, critical industrial infrastructure, vertical sectors and other new network services. However, these new network features and the wide application of 6G are fraught with potential security and privacy challenges. In addition, the OS architecture means that based on general-purpose hardware, any potential user is able to study, update and release new software versions to other users in a collaborative open-access manner. This however highlights the virtualization security issues, and poses a challenge in terms of the definition and protection of intellectual property. Therefore, how to effectively protect cybersecurity and intellectual property is still a huge challenge.

E. ENGAGING TELECOM VENDORS

Although more and more researchers and practitioners rely on open-source components to build customized cellular networks supporting MEC functionality, telecom vendors are not overly motivated to follow this approach. Since this open-source mode makes the specialized hardware associated with particular operating systems or functions in the original traditional network obsolete, vendors can no longer make profits by developing different specialized hardware. Therefore, how to persuade telecom vendors to join this open-source movement from an economic perspective and to further expand the target group supporting this movement is still a challenge.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we have conceived the OS-MEC scheme for 6G relying on sophisticated decoupling and reconfiguration principles. First of all, an OS-MEC framework was presented based on ETSI's MEC architecture. Then, by means of the SBA and NFV technology, the tightly coupled service functions were decomposed into independent NFs. Next, by introducing the template and instantiation based concepts into MANO, the disaggregated NFs may be re-assembled and the required resources may be allocated in order to provide customized services for users. In this context, the evolved NFs were illustrated in the service-based MEC layer. The key components and the associated implementation processes were described in the NF's MANO. Then, a pair of typical use cases have been critically appraised through demonstrations in a small-scale test network. Finally, we have discussed some potential technical challenges of 6G-oriented OS cellular networks.

In our future work, we will aim for developing the OS RAN architecture in order to construct a complete OS cellular network for 6G. Based on the twinned core principles of the OS RAN architecture: RAN decoupling and RAN reconfiguration, we envisage to propose a new three-dimensional RAN slicing mechanism, which realizes the decoupling and re-encapsulation of resources and NFs from the three dimensions of resource virtualization, function virtualization and network slicing, so as to provide customized virtual RANs for users. We hope to fill the gaps in the research of OS RAN architecture and lay the foundations for the ultimate realization of cloud-network-edge-terminal collaborative OS cellular networks.

LIST OF ABBREVIATIONS

3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5GC	5G Core Network
6G	Sixth Generation
AP	Access Point
API	Application Program Interface
APP	Application
ASF	Application Selection Function
CBS	Control Base Station
CDUS	Central Unit and Distributed Unit Splitting
CN	Core Network
COTS	Commercial Off-The-Shelf
CP	Control Plane
CPCF	Communication Protocol Conversion Function
CPU	Central Processing Unit
CUPS	Control Plane and User Plane Splitting
DBS	Data Base Station
DL	Downlink
DUDe	Downlink and Uplink Decoupling
EBI	East-Bound Interface
eNB	evolved Node Base Station
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FTP	File Transfer Protocol
gNB	Next-generation Node Base Station
HDD	Hard Disk Drive
HSDe	Hardware/Software Decoupling
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
LF	Linux Foundation
MANO	Management and Orchestration
MBS	Macro Base Station
MEC	Multi-access Edge Computing
MME	Mobility Management Entity
NBI	North-Bound Interface
NDN-ECC	Named Data Networking-based Edge Cloud Computing
NF	Network Function
NFV	Network Function Virtualization
NR	New Radio
NRF	NF Repository Function
OAI	Open-Air-Interface
ONAP	Open Network Automation Platform
OS-MEC	Open-source MEC
OPNFV	Open Platform of NFV
OS	Open-Source
PGW	Packet Data Network Gateway
RAM	Random Access Memory

RAN   Radio Access Networks
RESTful   Representational State Transfer-ful
SBA   Service-Based Architecture
SBI   Service-Based Interface
SBS   Small Base Station
SDEC   Software-Defined Edge Computing
SDN   Software-Defined Networking
SGW   Serving Gateway
SRF   Service Registry Function
SSD   Solid State Drive
THz   Terahertz
TN   Transport Network
UDM   Unified Data Management
UHD   Ultra High Definition
UL   Uplink
UP   User Plane
UPF   User Plane Function
URLLC   Ultra Reliable Low Latency Communications
vEPC   virtualized EPC
VIM   Virtualized Infrastructure Manager
VM   Virtual Machine
vNF   vitual NF

REFERENCES

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[2] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[3] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[4] *Mobile Edge Computing: A Key Technology Towards 5G*, White Paper, ETSI, Sophia Antipolis, France, Sep. 2015.

[5] A. Ksentini and P. A. Frangoudis, "On extending ETSI MEC to support Lora for efficient IoT application deployment at the edge," *IEEE Commun. Standards Mag.*, vol. 4, no. 2, pp. 57–63, Jun. 2020.

[6] G. Avino, P. Bande, P. A. Frangoudis, C. Vitale, C. Casetti, C. F. Chiasserini, K. Gebru, A. Ksentini, and G. Zennaro, "A MEC-based extended virtual sensing for automotive services," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1450–1463, Dec. 2019.

[7] S.-R. Yang, Y.-J. Tseng, C.-C. Huang, and W.-C. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/QoE models," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1888–1902, Feb. 2019.

[8] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies," *IEEE Access*, vol. 9, pp. 67512–67547, 2021.

[9] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[10] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[11] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5706–5712, Jul. 2020.

[12] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang, and M. Zhu, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, Jan. 2021, Art. no. 110301.

[13] C. Xu, N. Ishikawa, R. Rajashekar, S. Sugiura, R. G. Maunder, Z. Wang, L.-L. Yang, and L. Hanzo, "Sixty years of coherent versus non-coherent tradeoffs and the road from 5G to wireless futures," *IEEE Access*, vol. 7, pp. 178246–178299, 2019.

[14] S. Androutsellis-Theotokis, D. Spinellis, M. Kechagia, and G. Gousios, "Open source software: A survey from 10,000 feet," *Found. Trends Technol., Inf. Oper. Manage.*, vol. 4, nos. 3–4, pp. 187–347, 2011, doi: 10.1561/0200000026.

[15] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, "Network functions virtualization: The long road to commercial deployments," *IEEE Access*, vol. 7, pp. 60439–60464, 2019.

[16] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, and H. L. Owen, "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25487–25526, 2017.

[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, Nov. 2019.

[18] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: A review," *IEEE Access*, vol. 9, pp. 18706–18721, 2021.

[19] *The Status of Open Source for 5G*, White Paper: 5G Americas, Feb. 2019. [Online]. Available: https://www.5gamericas.org/the-status-of-open-source-for-5g/

[20] C.-L. I, S. Kuklinski, T. Chen, and L. Ladid, "A perspective of O-RAN integration with MEC, SON, and network slicing in the 5G era," *IEEE Netw.*, vol. 34, no. 6, pp. 3–4, Nov. 2020.

[21] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[22] *System Architecture for the 5G System; Stage 2*, document TS 23.501, 3GPP, 2018.

[23] H. C. Rudolph, A. Kunz, L. L. Iacono, and H. V. Nguyen, "Security challenges of the 3GPP 5G service based architecture," *IEEE Commun. Standards Mag.*, vol. 3, no. 1, pp. 60–65, Mar. 2019.

[24] W. Wang and H. Li, "Light-weight platform for attack validation in LTE network," *IEEE Netw. Lett.*, vol. 2, no. 4, pp. 212–215, Dec. 2020.

[25] P. K. Taksande, P. Jha, A. Karandikar, and P. Chaporkar, "Open5G: A software-defined networking protocol for 5G multi-RAT wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2020, pp. 1–6.

[26] S. Lagen, L. Giupponi, A. Hansson, and X. Gelabert, "Modulation compression in next generation RAN: Air interface and fronthaul trade-offs," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 89–95, Jan. 2021.

[27] M. Pattaranantakul, R. He, Z. Zhang, A. Meddahi, and P. Wang, "Leveraging network functions virtualization orchestrators to achieve software-defined access control in the clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 372–383, Jan. 2021.

[28] A. Boubendir, E. Bertin, and N. Simoni, "On-demand dynamic network service deployment over NaaS architecture," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 1023–1024.

[29] P. Hu, W. Chen, C. He, Y. Li, and H. Ning, "Software-defined edge computing (SDEC): Principle, open IoT system architecture, applications, and challenges," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5934–5945, Jul. 2020.

[30] R. Ullah, M. A. U. Rehman, and B.-S. Kim, "Design and implementation of an open source framework and prototype for named data networking-based edge cloud computing system," *IEEE Access*, vol. 7, pp. 57741–57759, 2019.

[31] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[32] L. Yan, X. Fang, and Y. Fang, "Stable beamforming with low overhead for C/U-plane decoupled HSR wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6075–6086, Jul. 2018.

[33] H. Gupta, M. Sharma, A. F. A., and B. R. Tamma, "Apt-RAN: A flexible split-based 5G RAN to minimize energy consumption and handovers," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 473–487, Mar. 2020.

[34] M. A. Lema, E. Pardo, O. Galinina, S. Andreev, and M. Dohler, "Flexible dual-connectivity spectrum aggregation for decoupled uplink and downlink access in 5G heterogeneous systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 2851–2865, Nov. 2016.

[35] A. Ksentini and P. A. Frangoudis, "Toward slicing-enabled multi-access edge computing in 5G," *IEEE Netw.*, vol. 34, no. 2, pp. 99–105, Mar. 2020.

[36] A. J. Gonzalez, G. Nencioni, A. Kamisiński, B. E. Helvik, and P. E. Heegaard, "Dependability of the NFV orchestrator: State of the art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3307–3329, 4th Quart., 2018.

[37] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020.

[38] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[39] S. Rautmare and D. M. Bhalerao, "MySQL and NoSQL database comparison for IoT application," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ICACA)*, Oct. 2016, pp. 235–238.

[40] K. Kaur, S. Garg, G. Kaddoum, S. H. Ahmed, and M. Atiquzzaman, "KEIDS: Kubernetes-based energy and interference driven scheduler for industrial IoT in edge-cloud ecosystem," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4228–4237, May 2020.

[41] T. Li, L. Zhao, R. Duan, and H. Tian, "SBA-based mobile edge computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[42] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar./Apr. 2021.

[43] Q. Xia, Z. Hossain, M. Medley, and J. M. Jornet, "A link-layer synchronization and medium access control protocol for terahertz-band communication networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 2–18, Jan. 2021.
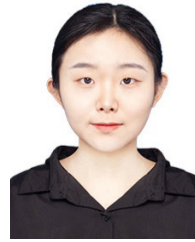
GUORONG ZHOU is currently pursuing the Ph.D. degree with the School of Telecommunication Engineering, Xidian University, China.

Since April 2021, she has been a Visiting Ph.D. Student at the School of Electronics and Computer Science, University of Southampton, Southampton, U.K., on the funds from the Program of the China Scholarships Council. Her research interests include wireless communications technologies, network slicing, multi-access edge computing, and machine learning for communications.

GAN ZHENG (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in electronic and information engineering from Tianjin University, Tianjin, China, in 2002 and 2004, respectively, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, in 2008. He is currently a Reader of signal processing for wireless communications with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, U.K. His research interests include machine learning for communications, UAV communications, mobile edge caching, full-duplex radio, and wireless power transfer. He was a recipient of the 2013 IEEE Signal Processing Letters Best Paper Award. He also received the 2015 GLOBECOM Best Paper Award and the 2018 IEEE Technical Committee on Green Communications & Computing Best Paper Award. He was listed as a Highly Cited Researcher by Thomson Reuters/Clarivate Analytics in 2019. He also serves as an Associate Editor for the IEEE Communications Letters and the IEEE Wireless Communications Letters.

LIQIANG ZHAO (Member, IEEE) received the B.Sc. degree in electrical engineering from Shanghai Jiao Tong University, China, in 1992, and the M.Sc. degree in communications and information systems and the Ph.D. degree in information and communications engineering from Xidian University, China, in 2000 and 2003, respectively.

From 1992 to 2005, he was a Research Engineer with the 20th Research Institute, Chinese Electronics Technology Group Corporation (CETC), China. From 2005 to 2007, he was an Associate Professor with the State Key Laboratory of Integrated Service Networks (ISN), Xidian University. He was appointed as a Marie Curie Research Fellow at the Centre for Wireless Network Design (CWiND), University of Bedfordshire, in June 2007, to conduct research in the GAWIND project funded under EU FP6 HRM Program. His activities focused on the area of automatic wireless broadband access network planning and optimization. Since June 2008, he has returned Xidian University, first as an Associate Professor and later as a Professor. He has hosted/participated many national research projects, such as the National Natural Science Foundation, the 863 Program, and the National Science Technology Major Projects, and several international research projects, including the EU FP6, FP7 plans for international cooperation exchange projects, and some research projects from companies, such as Huawei. He has more than 100 published in authorized academic periodicals both in and abroad and in international science conferences, wherein 30 of which are retrieved in SCI, and more than 70 of them are EI indexed, and six national invention patents. His current research interests include mobile communication systems, spread spectrum communications, WiMAX, WLAN, wireless sensor networks, broadband wireless communications, and space communications. In 2008, he was awarded by the Program for New Century Excellent Talents in University, Ministry of Education, China.

CHIH-LIN I (Fellow, IEEE) received the Ph.D. degree in EE from Stanford University.

She is currently a CMCC Chief Scientist of wireless technologies. She has published over 200 papers in scientific journals, book chapters, and conferences. She holds over 100 patents. She is the coauthor of the book *Green and Software-defined Wireless Networks: From Theory to Practice* and has also co-edited two books: *Ultra-dense Networks: Principles and Applications* and *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*. Her current research interests include ICDT deep convergence: from green and soft to open and smart.

Dr. I is a fellow of WWRF. She is the Chair of an O-RAN Technical Steering Committee and an O-RAN Executive Committee Member, the Chair of FuTURE 5G/6G SIG, the Chair of WAIA (Wireless AI Alliance) Executive Committee, an Executive Board Member of GreenTouch, a Network Operator Council Founding Member of ETSI NFV, a Steering Board Member and the Vice Chair of WWRF, a Steering Committee Member and the Publication Chair of the IEEE 5G and Future Networks Initiatives, the Founding Chair of the IEEE WCNC Steering Committee, the Director of IEEE ComSoc Meetings and Conferences Board, a Senior Editor of the IEEE Transactions on Green Communications and Networking, a member of IEEE ComSoc SDB, SPC, and CSCN-SC, and a Scientific Advisory Board Member of Singapore NRF. She has won the 2005 IEEE ComSoc Stephen Rice Prize, the 2018 IEEE ComSoc Fred W. Ellersick Prize, the 7th IEEE Asia-Pacific Outstanding Paper Award, and the 2015 IEEE Industrial Innovation Award for Leadership and Innovation in Next-Generation Cellular Wireless Networks.

XIAOHU YOU (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Nanjing Institute of Technology, Nanjing, China, in 1982, 1985, and 1989, respectively. From 1987 to 1989, he was a Lecturer with the Nanjing Institute of Technology. Since 1990, he has been with Southeast University, first as an Associate Professor and later as a Professor. He is the Chief of the Technical Group of China 3G/B3G Mobile Communication Research and Development Project. His research interests include mobile communications, adaptive signal processing, and artificial neural networks with applications to communications and biomedical engineering. He received the Excellent Paper Prize from the China Institute of Communications in 1987 and the Elite Outstanding Young Teacher Awards from Southeast University in 1990, 1991, and 1993. He was also a recipient of the 1989 Young Teacher Award of Fok Ying Tung Education Foundation, State Education Commission of China.

LAJOS HANZO (Life Fellow, IEEE) received the master's and Ph.D. degrees from the Technical University (TU) of Budapest, in 1976 and 1983, respectively, and the D.Sc. degree from the University of Southampton, in 2004. He has published more than 1900 contributions at IEEE Xplore and 19 Wiley-IEEE Press books, and has helped the Fast-Track Career of 123 Ph.D. students. Over 40 of them are professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is a fellow of the Royal Academy of Engineering, IET, and EURASIP. He is a Foreign Member of the Hungarian Academy of Sciences and the former Editor-in-Chief of the IEEE Press. He has served several terms as the Governor of both IEEE ComSoc and VTS. He was awarded the Honorary Doctorates by the TU of Budapest, in 2009, and by the University of Edinburgh, in 2015.

• • •