IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 18, 2020, accepted October 20, 2020, date of publication October 27, 2020,
date of current version November 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034136

Multi-Access Edge Computing: A Survey

ABDERRAHIME FILALI[1], AMINE ABOUAOMAR[1,2], (Student Member, IEEE), SOUMAYA CHERKAOUI[1], (Senior Member, IEEE), ABDELLATIF KOBBANE[2], (Senior Member, IEEE), AND MOHSEN GUIZANI[3], (Fellow, IEEE)

[1]INTERLAB, Engineering Faculty, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada
[2]ENSIAS, Mohammed V University, Rabat 10056, Morocco
[3]Qatar University, Doha, Qatar

Corresponding author: Abderrahime Filali (abderrahime.filali@usherbrooke.ca)

ABSTRACT Multi-access Edge Computing (MEC) is a key solution that enables operators to open their networks to new services and IT ecosystems to leverage edge-cloud benefits in their networks and systems. Located in close proximity from the end users and connected devices, MEC provides extremely low latency and high bandwidth while always enabling applications to leverage cloud capabilities as necessary. In this article, we illustrate the integration of MEC into a current mobile networks' architecture as well as the transition mechanisms to migrate into a standard 5G network architecture. We also discuss SDN, NFV, SFC and network slicing as MEC enablers. Then, we provide a state-of-the-art study on the different approaches that optimize the MEC resources and its QoS parameters. In this regard, we classify these approaches based on the optimized resources and QoS parameters (i.e., processing, storage, memory, bandwidth, energy and latency). Finally, we propose an architectural framework for a MEC-NFV environment based on the standard SDN architecture.

INDEX TERMS 5G, multi-access edge computing, network function virtualization, network slicing, service function chaining, software defined networking.

I. INTRODUCTION

In the last few years, the world has become more connected on a much deeper level, which is affecting the amount of data exchanged between the different actors of the network. The 5G specifications promise to significantly improve computing, storage, and network performance in different use cases. This can be seen in various industries such as vehicles which will be able to provide information related to roads using the various on-board sensors which capture and report data in real time. Also, 5G has the potential of reinventing agriculture through smart farming that depends on dedicated equipment with processing and networking capabilities to achieve real-time, precise production and management [1], [2]. In addition, 5G allows connections between distributed cloud networks in different geographic regions. As a result, telecommunication and IT ecosystems, including infrastructure providers, service providers and Over-the-Top (OTT) providers are in full technological transformation.

The new generation of applications produces a huge amount of data and requires a variety of services, which fuels the need for extreme network capabilities in terms of ultra-low latency, high bandwidth and resource consumption. These requirements are the reason why the telecommunication and IT ecosystems are progressively trending toward exploiting Multi-access Edge Computing (MEC) paradigm to improve the provided services and reduce OPEX/CAPEX. MEC consists in moving the different resources from distant centralized cloud infrastructure to edge infrastructure closer to where the data is produced. In fact, instead of offloading all the data to be processed on a cloud infrastructure, edge networks act as mini datacenters that analyze, process and store the data. Accordingly, MEC reduces latency and provides high-bandwidth applications with real-time performance [3]. Processing data at the edge of the networks has other important advantages, such as minimizing the risk of traffic congestion and decreasing data transmission costs. Consequently, MEC is a major enabler to achieve 5G objectives, which include supporting enhanced Mobile Broad Band (eMBB), Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine-Type Communications

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

(mMTC). The applications of MEC are expanding, so following this trend, several standardization initiatives are being conducted to ensure a successful integration of MEC into cellular and non-cellular networks [4]. Among these initiatives, the most prominent projects are, ETSI-MEC [5], Edge Computing Consortium [6], Open Edge Computing Initiative [7] and Central Office Re-architected as a Data Center (CORD) [8]. For instance, the ETSI MEC industry standardization group aims to define the necessary specifications for an open and standardized environment that helps multi-vendor MEC platforms to integrate applications from interested parties in delivering MEC-based services. MEC incorporation into cellular and non-cellular networks will be driven by the influence of Software-Defined Networking (SDN) [9], Network Function Virtualization (NFV) [10], Service Function Chaining (SFC) [11] and Network Slicing [12]. With SDN, MEC environment management becomes more flexible, programmable and consolidated to define where and how data is processed. NFV promises many benefits to the MEC environment, including flexible provisioning (i.e., scale up/down) of computing and storage resources, fast deployment of new services and reduction of hardware costs through virtualization. MEC can leverage SFC concepts for optimizing the network resource utilization and enhancing application performance. MEC and network slicing can be used together to enable operators to deploy networking platforms on demand. This fruitful cooperation between MEC and the aforementioned paradigms (i.e., SDN, NFV, SFC and network slicing) is foreseen in the 5G network in addition to others non-cellular network.

In recent years, MEC paradigm has attracted a great interest from both academia and industry researchers and many surveys have been published [13]–[20]. The work of [13] presents the MEC paradigm as a 5G technology taking into account ETSI reference architectures and recommendations. The work of [14] proposes a comprehensive survey of the associativity between MEC and IoT technologies. Authors of [15] investigate computational tasks offloading to the MEC by end-users. In [16], the focus was on the joint radio and computational resource allocation for MEC. The work of [17] highlights the game theory models that have been applied to MEC environments and discusses the benefits of game theory as a powerful framework allowing MEC to deal with new use case requirements. A state-of-the-art of MEC and an overview on future research directions have been introduced in [18]–[20]. The work of [19] investigates MEC within an SDN context and shows how MEC could leverage SDN to improve its performance. It also discusses the cooperation between MEC and SDN as 5G enablers. In contrast to the existing surveys on MEC, the fundamental objective of our work is to provide a detailed overview on optimization approaches for MEC environment from the architectural perspective and from the relationship between MEC and other 5G enabling technologies, namely NFV, SDN, SFC and network slicing. We also discuss the relevant optimization approaches that aim to ensure a good quality of service within a MEC infrastructure and we classify these works based on the type of resource (i.e. energy, processing, storage/memory and bandwidth), the QoS parameter (i.e. latency) and the used optimization approach.

This survey is a valuable addition to existing works and can help readers to acquire a solid knowledge of a various approaches that optimize the MEC resources and its QoS parameters. Furthermore, we propose an architectural framework that combines the MEC and NFV technologies based on the SDN architecture. This framework illustrates the best placement of the SDN controller to perform its main orchestration and management role in the MEC-NFV environment. The main contributions of this work can be summarized as follows:

- We highlight the different options to integrate MEC into the current standard cellular network architectures and the different ways to perform a migration to the 5G network architecture.
- We elaborate the impact of the different technologies related to MEC, namely NFV, SDN, SFC and network slicing.
- We propose a description of MEC environment optimization approaches and a classification of these approaches according to the optimized resource and QoS parameters.
- We propose a MEC-NFV architectural framework based on the SDN architecture that ensures a better network performance.

The rest of this article is structured as follows. Section II provides a description of the different ways to integrate MEC paradigm into current cellular network architectures and the transition mechanisms for the 5G architecture. Section III defines the concepts of NFV, SDN, SFC and network slicing technologies and highlights their benefits as MEC enablers. Section IV provides a comprehensive review for the most relevant optimization approaches considering the optimized resource. Section V describes the proposed MEC-NFV architectural framework based on SDN architecture. The paper is concluded in Section VI. For the sake of clarity, Fig. 1 shows the organization of this article.

II. MEC INTEGRATION INTO 5G

MEC is considered as a key enabler that allows operators to integrate application-oriented capabilities to their network. This integration will allow operators and service providers to cope with cases where latency is critical [21]. MEC deployment could be performed across many scenarios where the network architecture and generation has absolutely no impact on the deployment. It is important to note that MEC technology is not limited to 5G, but it is a key feature to enable the 5G and make it possible. MEC as a universal access technology that enables low latency requirements whenever it is needed in scenarios where the locality is required such as autonomous vehicles. In what will follow, we introduce the different architectures proposed by ETSI to integrate the

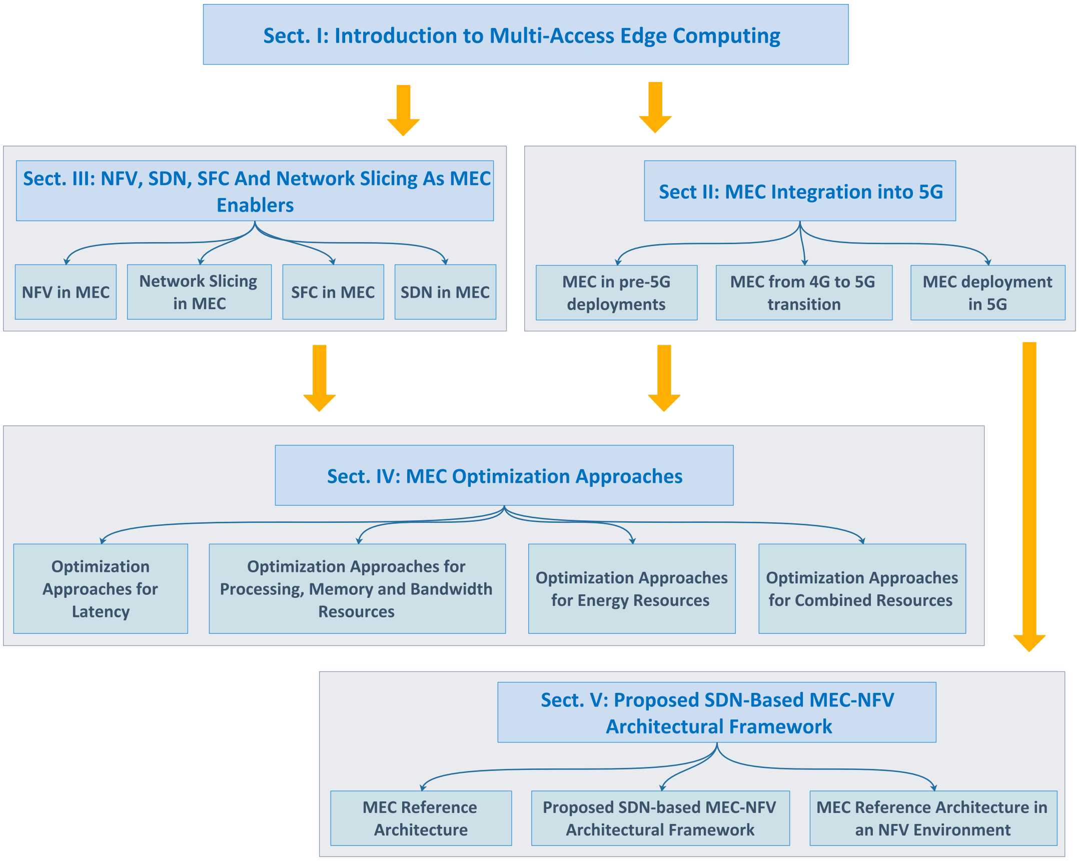

Sect. I: Introduction to Multi-Access Edge Computing
Sect. III: NFV, SDN, SFC And Network Slicing As MEC Enablers
NFV in MEC
Network Slicing in MEC
SFC in MEC
SDN in MEC
Sect II: MEC Integration into 5G
MEC in pre-5G deployments
MEC from 4G to 5G transition
MEC deployment in 5G
Sect. IV: MEC Optimization Approaches
Optimization Approaches for Latency
Optimization Approaches for Processing, Memory and Bandwidth Resources
Optimization Approaches for Energy Resources
Optimization Approaches for Combined Resources
Sect. V: Proposed SDN-Based MEC-NFV Architectural Framework
MEC Reference Architecture
Proposed SDN-based MEC-NFV Architectural Framework
MEC Reference Architecture in an NFV Environment

FIGURE 1. Structure of the paper.

MEC into pre-5G technologies and how the migration could be performed to the 5G architecture.

A. MEC IN PRE-5G DEPLOYMENTS

One of the main functionalities of MEC platform is the packets' routing to MEC applications which could be performed either in the breakout mode, inline mode, tap mode or in the independent mode. In the breakout mode, the session connection is being redirected to a locally hosted MEC application or to a remote one [22]. An example of a breakout application is the local CDN or an enterprise LAN. In the inline mode, the session, unlike the breakout mode, is being maintained with the original server (through Internet) while all the traffic pass through the MEC application. For instance, caching and security applications are considered as inline applications. The tap mode, the traffic is replicated and being forwarded to the MEC application, such as deploying a virtual network of security applications. Finally, the independent mode, as its name indicates, the traffic offloading is not required, yet the application is still registered to the MEC infrastructure and could process MEC services such as DNS. The MEC deployment in a pre-5G cellular architecture could be performed through different ways and manners: (i) Bump in the wire, (ii) Distributed EPC, (iii) Distributed S/PGW and (iv) Distributed SGW with Local Breakout (SGW-LBO).

1) BUMP TO THE WIRE

As its name indicates, bump in the wire includes all the cases where the MEC infrastructure deployment takes place between the base station and the core network. Fig. 2 illustrates an example of bump to the wire deployment. A MEC platform is bundled by the eNodeB to act as a single entity to give it the possibility to route plain IP packets to and from the MEC applications. Also, routing the GPRS Tunneling Protocol (GTP)-encapsulated packets to and from the serving gateway (SGW) for usual traffic through well-known modes such as S1-U interface. Such deployment could be adapted by enterprises in order to allow the intranet traffic to break out to local applications similar to local IP access and in scenarios where the MEC platform is deployed aside with a CRAN deployment as proposed in [22].

The other deployments of MEC in the pre-5G cellular networks are either deployed in proximity of the radio node or at an aggregation point on the S1 interface. In such deployment, the data plane of the MEC's hosts has to process the GTP-encapsulated traffic and in some cases, a portion of the data plane that could be generated locally at the MEC's host or could come from a local breakout without passing through the core network. As a solution, a Control/User Plane Separation could be implemented by deploying a gateway to be in charge of intercepting and processing such traffic as

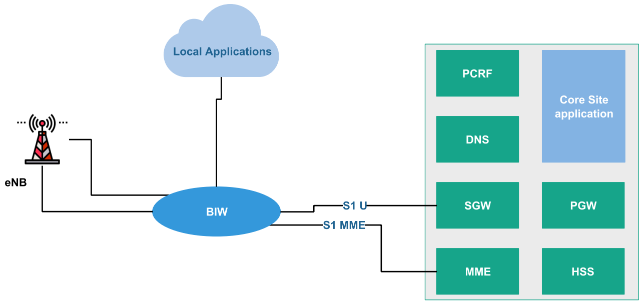


FIGURE 2. Bump in the wire deployment option.

depicted in Fig. 2 (MEC GW). This solution ensures in the first-place low latency since we have the possibility to deploy MEC platforms between the core network and the eNodeB. It also ensures the traffic steering in both session-oriented steering and traffic granularity steering.

BIW deployment still suffers from some limitations, such as security, reachability and traffic charging. The IPsec is used to secure the S1 interface between the eNodeBs and the core network which enables traffic interception. Therefore, to allow the BIW entity to intercept traffic, IPsec needs to be disabled or being limited to somewhere behind the IPsec gateway where the messages could be intercepted clearly (no encryption). In such deployment, the operator is limited from the perspective the MEC's placement which reduces the distributivity of the MEC platform. Consequently, MEC platform may not respect the factor of proximity which consists in being as close as possible to UEs. Considering the reachability of idle users, adopting BIW option will increase the delays to the initiation phase of the connection which is not practical. In addition, applications will not be able to setup connections towards devices in power-saving (IDLE) mode. This is because a BIW solution does not support the set of functionalities required to wake up a device in IDLE mode. It may achieve that, where possible, only by relying on a PGW node that will be at a different location thus introducing latency. From the charging perspective, Charging Data Record (CDR) for steered traffic is difficult to be produced in a BIW deployment option due to the fact that the MEC platform does not have access to all necessary information (IMSI, IMEI, IP address, etc.) to produce CDRs. Therefore, charging could be only done by adding complexities (e.g. new non-standard 3GPP network functions and interfaces) into the operators' network [23]. As a solution to these limitations, operators need to deploy additional equipment, which could not be optimal from the cost point of view. Also, adding equipment could make the architecture of the network a bit more complex which may decrease the efficiency of the deployment [24].

2) DISTRIBUTED EPC (DEPC)

In a DEPC deployment, the MEC host includes all the parts of 3GPP specified by 4G systems described in [25]. In this deployment model, the MEC data plane lies on the SGi interface. To steer the traffic towards the MEC platform, MEC's local DNS and PDN Gateways (PGW) are the essential components of a DEPC. When a user subscribes to a DEPC, the MEC's applications IP address resolving is performed by the PGW through local DNS interrogation. PGW also have the role of terminating the PDN connections and assigning the IP addresses. In such deployment, it is very important to keep a stable network architecture, since the 3GPP interfaces are used for other operations such as session management.

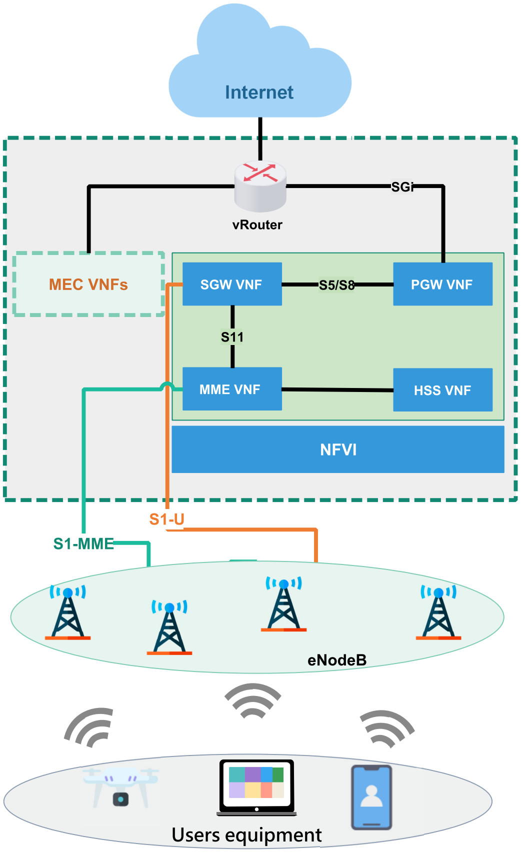


FIGURE 3. Distributed EPC MEC deployment.

Such deployment is also considered to be suitable for mission critical push to talk (MCPTT) scenarios and M2M scenarios where the communication between the UE and the core network is not required [22]. Fig. 3 illustrates the different components of a DEPC deployment option of MEC. In this deployment option, the Home Subscriber Server (HSS) is deployed as a VNF on the edge platform aside with the other EPC components. In fact, the HSS is managed in a centralized fashion by the operator at the core side. In this case, using a backhaul connection is not required to keep the servers sunning. The benefit of such deployment is to enable and allow the local management of the subscriber's database and taking profits of local EPC in MEC to offload the entire traffic of the APN. In addition, the DEPC is more suitable in scenarios where the operator is interested in delivering the exact QoS and configurability features.

A special case of the DEPC is the distributed S/PGW deployment. The difference between DEPC and S/PGW resides at the SGW and PGW entities, which consists in deploying it at the edge platform, while the control plane is located at the operator's premises and the data plane of the MEC is connected to the PGW entity through the SGi. Similar to the previous option with the fully distributed EPC, the SGW and PGW could run as VNFs [22]–[26] alongside with the MEC applications on the NFV infrastructure within the same MEC host [23]. The selection of SGW is delegated to the central MME based on tracking area code of the radio interface to which the user is associated. This deployment is very useful in scenarios where the traffic offloading is considered. This architecture allows to offload the traffic based on the APN, which means that the IMS for VoLTE APN and roaming APNs may not be offloaded. In addition, the SGW and PGW entities are deployed at the same network edge. Such deployment requires an extension of the S5 interface to the MEC site. This deployment allows the operator to fully control the MME.

3) DSGW WITH LOCAL BREAKOUT

Local breakout at the SGWs is a new architecture for MEC that originates from operators' desire to have a greater control on the granularity of the traffic that needs to be steered [22]–[24]. This deployment mode offers the operator a better control on the granularity of the steered traffic. Also, it gives the users more flexibility to access both MEC hosted applications and the core network through the same APN. In some cases, DSGW deployment offers the possibility to deploy MEC hosts alongside with the SGW, thus, the MEC application could be implemented as VNFs within the same MEC infrastructure. To steer the traffic, the SGi-enabled local breakout is used. This interface allows the traffic splitting and the high level of security such as 3GPP security requirements.

Such deployment enables the operators to have a control on traffic filters, such as uplink classifiers of 5G communication networks used to steer the traffic. Therefore, it is possible with this deployment to steer the traffic based on steering policies of the operator. SGW-LBO also supports MEC host mobility and allows to deploy ultra-low latency applications.

Fig. 4 illustrates how to deploy MEC hosts alongside with the SGW in a mobile network, in which the MEC system and the DSGW are deployed at the edge. In this deployment, the MEC's SGW plays the main role and it is connected almost to every external entity: (i) core's PGW through the S5 interface based on GTPv1-C and GTPv1-U, (ii) core's MME through S11 (Green link in Fig. 4), (iii) eNodeBs through S1-U interface based on the GTPv1-U, and (iv) exchange the data with the MEC's platform through SGi-LBO interface using API based communication (LBO-API; gray link in Fig. 4). The billing system is connected to the CGF entity to enable the offline charging by fetching the CDR.

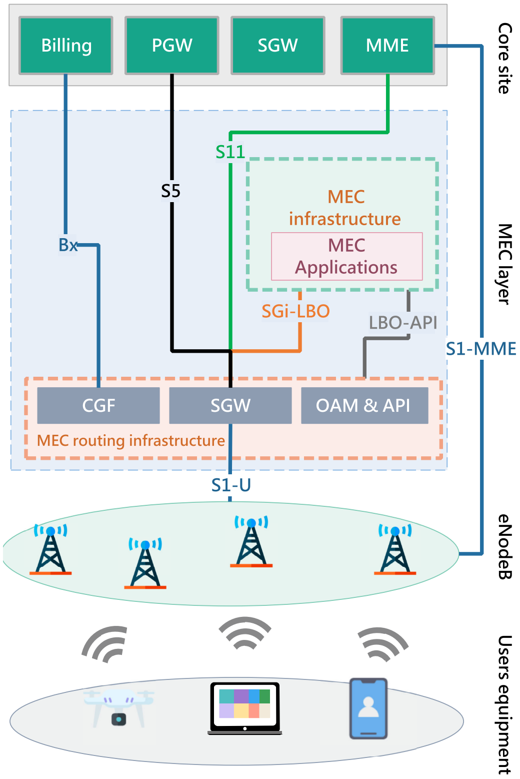


FIGURE 4. DSGW-LB MEC deployment.

B. MEC FROM 4G TO 5G TRANSITION

MEC has no constraints on the underlying delivery technology such as the radio interface, which makes it very flexible, and highly adaptive in the communication networks. In fact, the technology used in delivery, combined with the hardware of the MEC platform enables new forms of adaptability to the deployments. From the economic perspective, service providers (SPs) could leverage MEC to enhance their revenues through application production with no restriction to the network technology or generation (No need to wait for full deployment of the 5G and the associated capital investment). In fact, SP services will change to a whole new level where SPs will provide high quality of experience and services through the application's hosting in virtualized environment. To this end, MEC offers a smooth upgrade to the network technology (5G and 6G) with no need to major upgrade (i.e. in terms of hardware and software requirements). By enabling the virtualization technology, MEC will offer the operators more efficiency and accuracy in controlling the applications requirements in terms of resources. Therefore, enabling an efficient and accurate trade-off between the pricing and required resources of applications.

Coordination between the MEC's data plane and the 5G system's data plane is needed to forward and steer traffic to applications and network's data. For example, we could deploy an Application Function (AF) that interacts with the 5G control plane functions to enhance traffic routing and steering, collect information about 5G network capabilities and enable mobility. The 4G to 5G transition offers the

TABLE 1. Summary of MEC deployment options in the 4G architecture.

Deployment option	Summary
BIW	• Placed on the backhaul link of the base station. • Allows the interception of signaling and data traffic on S1 interface and steers it to the local MEC applications. • Traffic steering is based on configured policies. • BIW entity decides to steer some traffic out to applications outside the core network.
SGW	• Ensure that interfaces S11, S5 and Bx reach the Core Network through the MEC platform. • Ensure the S1-U network reachability on the RAN through the MEC platform. • Offloading the traffic. • Updating the operator's DNS to make MME select the MEC platform for the Tracking Area where the eNBs that need to be served are located. • The MEC applications are either hosted on the MEC platform or use through MEC APIs to communicate.
DEPC	• MEC includes logically all or part of the 3GPP Evolved Packet Core (EPC) components. • MEC data plane relay on the SGi interface. • UE subscribes to the distributed EPC co-located with the MEC host. • PGW terminates the PDN connection and assigns the IP address and local DNS information to resolve the MEC application's IP address. • Adequate for long term deployments. • Reduces costs as the EPC and its components can run as VNFs.

possibility to reuse the edge computing resources through an orchestration of the applications and the 5G network functions, at the same time MEC still orchestrate the application services. As an evolution of the current mobile networks, 5G allows an easy deployment of the data plane in order to enable edge computing. Also, 5G's service-based architecture [25] is composed of diverse control plane functional entities such as PCF, SMF and AF and data plane functional entities such as UPF. Therefore, MEC could be deployed flexibly and with the one defined for the 5G. An example of the deployment of MEC in 5G ecosystem is illustrated in Fig. 5, in which MEC's data plane is mapped to 5G's UPF and AF elements.

Fig. 6 shows an example of MEC deployment transition from the 4G deployment to the 5G. In such deployment, SGW-LBO deployment mode becomes a UPF-LBO, the links between the edge site and the core network becomes an N4 instead of S11 and N9 instead of S5. In addition, SGW-LBO is homogenous with 3GPP standards and could be mapped into 5G functionalities such as UPF and networks slicing.

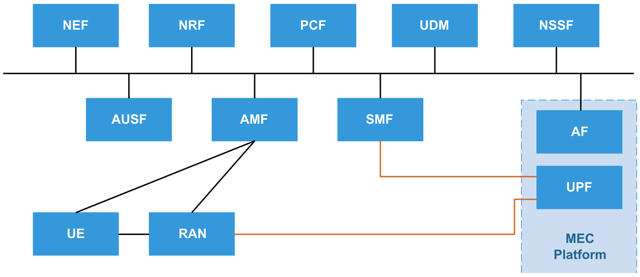


FIGURE 5. Overview of MEC deployment in 5G architecture.

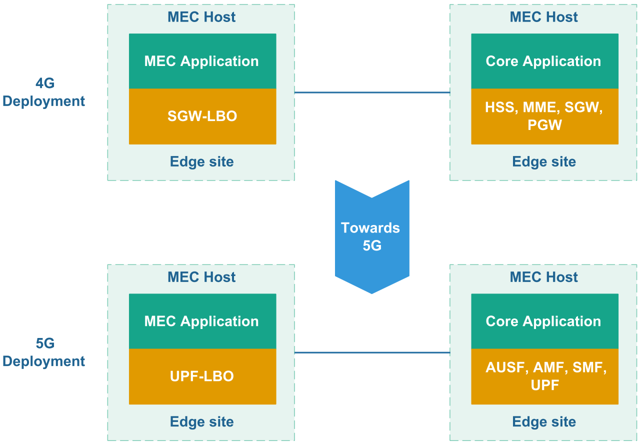


FIGURE 6. Example of 4G to 5G MEC's migration.

C. MEC DEPLOYMENT IN 5G

In its current deployment, MEC in the 4G architecture is linked to the user plane using the different deployment modes discussed in the previous sections. Therefore, MEC was designed to be a 4G add-on offering the possibility to deploy and place services at the edge of the network. MEC has been defined by ETSI in [27] as a huge extent self-contained, with full coverage and hand on everything starting from resource management and orchestration to the different interactions with the data plane for steering specific traffic. With 5G, MEC is defined as a key technology to enable low latency and mission-critical use cases for IoT services [21]. Consequently, the system is designed to enable high performance and quality of experience (QoE) by providing flexible and efficient support for edge computing. MEC could be mapped onto Application Function (AF) that could use services and metadata offered by other network entities and function based on a given policy to be configured by the supervising entity of the infrastructure.

In 5G Service Based Architecture (SBA) proposed by 3GPP, we can distinguish two kinds of functions: (i) those that consume one or many services and (ii) those that produce services. The exchange of services (produce/consume) is based on authentication mechanisms to grant authorization to the consumers. SBA allows flexibility and efficiency in service exposure. Some of the used methods are the request/response model for simple and lightweight service requests. For long-lived processes, a subscribe/notify model is supported by the

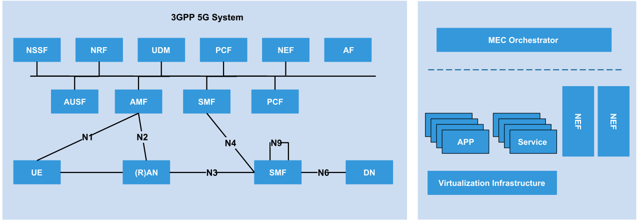


FIGURE 7. 5G service architecture (on the left), and a generic MEC deployment (on the right).

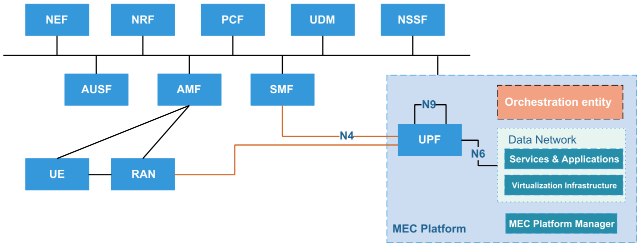


FIGURE 8. Integrated MEC deployment in a 5G architecture.

model to allow the efficiency in exchanging services and information between the entities. In [28] ETSI ISG MED proposes a full guideline to develop applications that support such functionalities and features. This proposed guideline offers the same features (registration, discovery, availability, registration/authentication and authorization, etc.) for MEC applications as the one offered by SBA for network functions and their services.

Fig. 8 illustrates the 3GPP 5G system [25] with its SBA on the left and the MEC system architecture on the right. The deployment of a MEC system in a 5G environment in an integrated manner requires from some of the functional MEC in 5G (blue boxes in MEC system part) to interact with the NFs of the 5G network. The network functions and the services are registered in the Network Resource Function (NRF) entity, while in MEC, the services produced by applications are registered in the service registry of the MEC platform. To use a service, an authorization is required from a network function in order to interact with the network function that produces the service. This kind of authorization is granted by the Authentication Server Function (AUSF). The discovery of the available services is proposed by the NRF. In some cases where the services are to be accessed by external and untrusted entities, NEF plays the same role as NRF. NEF could be seen as a centralized entity for service exposure to authorize all kinds of requests coming from outside of the system. In addition to AF, NEF and NRF, there are a number of other functions that are worth introducing. The PCF entity is charged to handle policies and rules such as traffic steering rules in a 5G network. The PCF could be accessed through NEF or directly depending on the trust degree of the AF. The Unified Data Management (UDM) function is responsible for many services related to users and subscriptions. It generates the credentials to authenticate users, handles user identification, access management (such as roaming), registers the user serving NFs (serving AMF, Session Management Function (SMF)), guarantee service continuity through history of SMF/Data Network Name (DNN) assignments. The User Plane Function (UPF) is considered as a distributed and configurable data plane from the MEC system point of view. Consequently, in some special scenarios of deployments, the local UPF could take part of the MEC deployment.

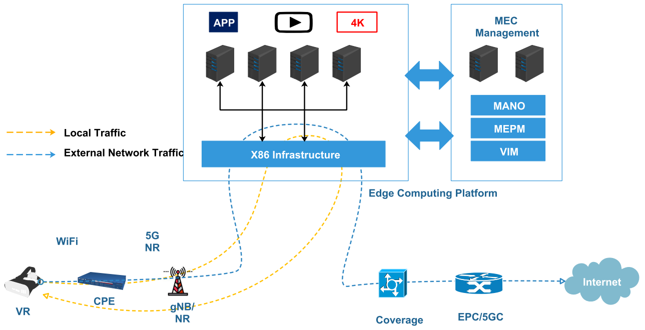


FIGURE 9. In-building streaming media system.

From the 5G architecture perspective, there is no implicit site selection for the MEC servers under resource amount (number of servers to deploy) and its cost. In addition, other challenges are to be considered such as the site selection and servers' density. However, independently from these constraints, MEC can be deployed on different levels of the 5G architecture, Namely on the: (i) wireless access side (RAN), (ii) edge datacenters outside the premises of the operator and (iii) core site of the network. From the RAN perspective, MEC servers could be deployed over eNodeBs, small base stations and even the operator's access points. However, in this deployment, the RAN should be open to independent service providers and not only the operators. At the edge datacenters, MEC can take place within the enterprise's sites over a set of servers and even end user equipment such as servers, laptops and mobile phones. On the core site, MEC can be deployed as described previously in this section.

On the right side of Fig. 9 we distinguish the MEC orchestrator of the MEC system which is a functional entity that acts as an AF. MEC orchestrator can interact with the NEF or NF depending on the use case. On the MEC host level it is the MEC platform that can interact with these 5G NFs as AFs. On the host level functional entities, the deployment is often taking place in the data network of a 5G system. While the NEF as a core network function is a system level entity deployed centrally together with a NFs, an instance of NEF could be deployed at the edge to enable low latency. As illustrated in Fig. 9, high-bandwidth applications, such as video conferencing, video streaming, and virtual/augmented reality suffer from the tradeoff between the network design and the application requirements. For example, traditional TCP congestion control is designed for wired networks perspectives and highly heterogeneous traffic. Cross-layer as considered as one of the most popular optimization tools across architectural boundaries which is considered as non-adequate for consumer products, but still, it is considered

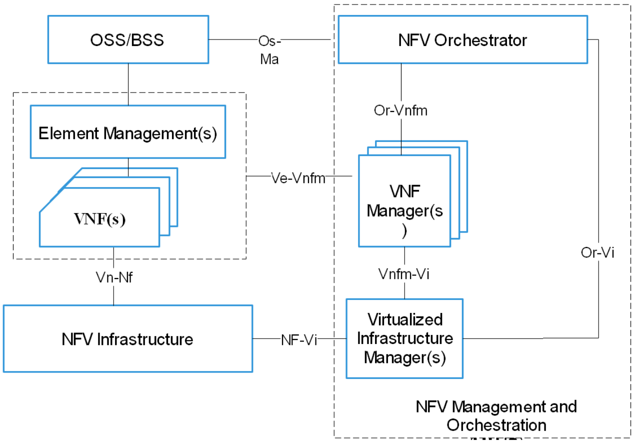


FIGURE 10. ETSI NFV reference architectural framework.

adequate and actually could bring significant and efficient network performance, resulting in both user QoE improvement and IT cost reductions [29].

III. NFV, SDN, SFC AND NETWORK SLICING AS MEC ENABLERS

In this section, we discuss the role of SDN, NFV, SFC and network slicing technologies as key pillars for MEC's operation in several aspects. For each technology, we define its characteristics and then we explain how the MEC can leverage these technologies to enhance the quality of the provided services.

A. NFV IN MEC

1) NFV DEFINITION

With the aim of delivering network services faster and avoiding the need to manually configure dedicated hardware devices to build service chains, operators and network service providers have been pushed to embrace NFV. The NFV concept provides a new way to abstract and virtualize the network functions, enabling them to be created, operated, distributed and controlled by software running on standard servers [10]. ETSI defined a list of use cases [30] that can be managed by NFV, namely virtualization of connectivity functions (e.g. DHCP, NAT router), mobile core network entities (e.g. MME, S/P-GW, HSS), RAN nodes functions (e.g. radio link control, packet data convergence protocol, radio resource control), network security functions (e.g. firewalls, deep packet inspection, intrusion prevention systems). NFV standardization and implementation efforts are driven by the ETSI Network Function Virtualization group, in collaboration with network operators and equipment vendors. NFV promises to bring agility in delivering network services and facilitate innovation in response to the explosive demands for new network services. In order to ensure reliability, scalability, legacy networks integration and connectivity with existing operational systems, ETSI designed a modular NFV architecture [31].

The main components of the NFV architecture, as illustrated in Fig. 10, include NFV Infrastructure (NFVI), Virtual Network Functions (VNFs) and NFV Management and Orchestration (MANO). VNF is responsible for providing a part or the whole network service and can be composed of several components knows as VNF Components (VNFCs). Thus, a VNF can be deployed in one or multiple virtual machines. Several VNFs can be ordered according to a certain logic in order to offer a particular network service or several network services. The NFVI is the combination of physical, virtual and software resources required to construct the environment in which the VNFs are hosted. Physical resources include computing, storage and networking components, which are virtualized using a hypervisor or a container management system. NFV MANO [32] presents the management and orchestration module of all these VNFs as well as the infrastructure on which they are deployed. The MANO consists of the Virtualized Infrastructure Manager (VIM), VNF Manager (VNFM) and the NFV Orchestrator (NFVO). The VIM allocates, restrains and handle virtualized resources dedicated to VNFs. The VNFM manages the lifecycle of VNFs and the Fault, Configuration, Accounting, Performance and security (FCAPS) of VNFs. Since there can be multiple VNFMs and VIMs in the system, the NFVO is responsible for the coordination between all these components as well as the management of end-to-end services among different VNFs.

2) NFV IN MEC

To achieve its objectives of high bandwidth, low latency and real-time access to the computing, storage and networking capabilities, MEC platform is designed to operate seamlessly within the NFV environment. This integration of the MEC into an NFV environment (e.g. deploying MEC applications and the MEC platform as VNFs and coexistence between MANO systems from MEC and NFV to manage their respective functions) is an effective way to take advantage of NFV concepts [33]. We list the following potential benefits of NFV to MEC: (i) reduction of CAPEX and OPEX; (ii) flexibility and fast deployment of new services.

a: REDUCTION OF CAPEX AND OPEX

MEC use cases (e.g. smart city services, IoT, connected vehicles, etc.) induce the appearance of multiple new applications and services. In order to meet this exponential increase of new network service requirements, operators and network service providers must continuously offer new network technologies, protocols, hardware-software solutions from different vendors. Therefore, capital and operational expenditures significantly increase, i.e. more physical space needed for network hardware, more network consumption and network maintenance costs will increase. To overcome this challenge, NFV provides network virtualization technology through migrating network functions from dedicated equipment to commodity hardware. NFV allows MEC services providers to expand and upgrade their environments with less cost.

For example, instead of installing a new hardware device to perform a network security function (e.g. firewall), a network administrator can apply the security function software on a standard server already activated in the network. The MEC and NFV frameworks have several similarities, particularly the MANO system characteristics. Hence, jointly manage MEC applications and virtualized network functions reduces CAPEX and OPEX expenses. The coexistence between MEC and NFV systems continues to attract many researchers [23], [34]–[37]. The work of [34] proposes a design for a common management and orchestration system in order to deliver end-to-end services that consist of MEC applications. The NFV-MANO system is extended to support a virtual application layer, which contains virtual application functions. Carella *et al.* in [37] propose a unified orchestration system that combines NFV and MEC use cases using NFV MANO Open Baton framework. The two NFV elements, VNFM and VIM, are customized to deal with the MEC requirements. They propose a new VIM driver to integrate Docker containers and a VNFM able to configure network services on top of containers. To avoid certain restrictions caused by the integration of MEC with the NFV architecture, authors in [23] present an effective framework. In this framework, NVFO and VNF managers are provided by Open Baton, while the MEC platform and MEC data plane are implemented by a distributed SGW with Local Breakout approach.

b: FLEXIBILITY AND FAST DEPLOYMENT OF NEW SERVICES

To meet the current and future requirements of network services, the MEC ecosystem must be flexible. NFV concepts provide to MEC an automatic configuration, upgrade and customization in order to accommodate changes in network services. Flexibility is achieved by the possibility of defining where, when and how many resources should be allocated among NFV-based MEC environment. Indeed, with the NFV capabilities, it is possible to place or migrate MEC service instances (i.e. VNFs) where they are needed and delete them if they are no longer needed. If a client desires a new service, the operator or service provider can instantiate a new VNF or multiple VNFs to perform the functions of that service. As easily as the service can be enabled, it can be disabled. Moreover, for a better flexibility, network services can be quickly scaled up or scaled down by provisioning the required resources to VNFs. Extensive studies on improving flexibility in NFV-enabled MEC are performed in [38]–[45]. For Ultra-Reliable Low-Latency Communications (uRLLC) services, authors in [38] propose a VNF placement scheme. Considering the very low latency required by uRLLC services, the algorithm tries to find a tradeoff between low service latency and high availability. To maximize the utilization of the MEC resources for low latency applications, a seamless application and VNF migration approach is presented in [39]. In this approach, if the NFV orchestrator accepts a request that has a predefined network and computation resource requirements, it determines where the accepted request should be executed and then migrates the application or the VNF. Also, a MEC-enabled testbed is introduced to verify the feasibility of the proposed approach. Cziva *et al.* [40] formulate a VNF placement optimization problem that minimizes the end-to-end latency of the MEC network. The proposed model is adapted to the changing network dynamics, user demand and mobility. First, they define an optimal placement problem of VNFs using an integer linear programming model. Then, in order to adapt to a real scenario (i.e. frequent latency changes and users' movements), they present a model to reschedule and recompute migration operations of VNFs. To meet the requirements of delay-sensitive applications in an NFV-based MEC environment, Yang *et al.* [41] have developed a set of algorithms for dynamically allocating resources. The proposed algorithms include: (1) an offline algorithm that allocates resources by making a tradeoff between operational costs and response time; (2) an online algorithm to scale up/down resources and balance the load considering workload fluctuations; (3) when the capacity of a node is exceeded by the second algorithm, a network latency constraint algorithm instantiates an edge node to support the required latency.

B. SDN IN MEC

1) SDN DEFINITION

To clearly define the SDN technology, it is necessary to introduce how a traditional network works and what are the symptoms that led to develop the SDN paradigm. In general, a network is composed of two main planes, which are the data plane and the control plane. The first one, also called the user plane or forwarding plane, presents the messages generated by the network users, which should be transferred according to a defined policy. In order to transfer these messages, the network must perform a lot of operations such as discovering the overall network topology, finding the shortest path and making decisions about where the traffic should be sent, etc. The exchanged requests to accomplish these operations present the control plane. This operating mode of the network makes its management and evolution difficult. For these reasons, SDN technology has been designed [9], [46].

SDN is developed to provide the flexibility that the control plane needs to support the traffic forwarding requirements of the data plane. According to the Open Network Foundation (ONF) [47], SDN is a dynamic architecture that promises to automate the network, centralize the control functions and program the network using APIs. Therefore, to achieve these objectives, SDN technology is based on three aspects: (1) separating the control plane and the data plane; (2) logically centralize the control plane [48]; (3) programing the control plane. According to the first aspect, the network intelligence is removed from the forwarding equipment and it is implemented into a logical instance called SDN controller. Thus, the operation of the data plane components becomes less complicated. The centralization of the control plane provides a global view of the entire network, which makes the configuration and the management of the network maintained and changed in a highly agile and adaptable way. Due to the centralization of the control plane, SDN controllers are now directly programmed through applications. This aspect has made it possible to develop new applications to better control and manage the network.

2) SDN IN MEC

ETSI propose a reference architecture to deploy the MEC system in an NFV environment [49]. Also, based on SDN architecture, several possible designs are proposed to establish a seamless cooperation between NFV and SDN technologies. Although these solutions have been designed for a reference NFV architecture, they remain available for a MEC-NFV environment. According to ETSI recommendations [32], multiple scenarios can be envisaged placing SDN controllers in a MEC-NFV architecture. An SDN controller can be located with the virtualized infrastructure manager, considered as part of the NFV infrastructure, virtualized as a VNF or merged with the OSS/BSS system. Therefore, SDN controllers are hosted in MEC servers to provide on-demand MEC services by connecting VNFs and dynamically manage the infrastructure resources (i.e. computing, storage and networking). MEC allows service providers to establish new types of services that required cloud computing capabilities at the edge of the network. SDN strengthens the MEC performance by addressing the importance of flexibility to define policies on how and where data is processed and to implement network services without additional investment and hardware modification [33]. We define five benefits that SDN brings to the MEC environment: (i) scalability; (ii) availability; (iii) resilience; (iv) interoperability; (v) extensibility. In the following, we discuss how the SDN can provide these benefits for a proper operation of MEC.

a: SCALABILITY

Scalability refers to the ability of a system to sustainably manage increased demands of its users. Accordingly, as the interest in IoT devices, connected vehicles and 5G NR-enabled applications increases, MEC environment must be scalable to suit the requested services by these applications. SDN enables MEC to support demand growth without disruption or redesigning of existing infrastructure. An SDN controller can manage the lifecycle of VNFs and services [50], i.e. instantiation, scale up, scale down and termination. In [51], an SND-based MEC framework is proposed and implemented, which provides scalability to PGW and SGW entities depending on the number of user equipment. A programmable QoE-SDN application is introduced in [52] to serve the customers of video streaming providers. This application allows video streaming providers to control the desired QoE by a direct communication with the mobile network operator. In a Vehicular Ad-Hoc Network (VANET), the fast movement of vehicles rapidly changes the topology of the network [53], [54]. These changes in the network topology generate more communications between vehicles and RSUs, vehicles and base stations, vehicles and MEC servers or RSUs and MEC servers. Thus, in order to increase the scalability of the MEC network to meet requested services by vehicles, SDN controllers are used in several works such as content delivery among connected vehicles [55], V2V data offloading [56] and the offloading of computation tasks from vehicles to the distributed edge servers [57].

b: AVAILABILITY

Availability is defined as the capacity of a system to perform the assigned functions properly whenever it is requested. For MEC technology, availability is the ongoing provision of the intended IT services to end-users and applications. MEC leverages the SDN paradigm for ensuring requested services availability respecting required performance in terms of low latency, high bandwidth and real-time access to the computing and storage capabilities available at the edge of the network. SDN approach makes the data plane components less complicated because their only task is forwarding data and sending information to one or several controllers [58]. Therefore, these components tend to have a higher availability. At the same time, receiving state information about the network components provides the controller with a global view of the whole network. Based on this global view, the SDN controller can decide to act either on the MEC system level (e.g. checking application rules and requirements and if necessary, adjust them to meet the required QoS) or on the MEC host level (e.g. configuring the infrastructure to satisfy the service level agreement) [59]. In an industrial IoT environment, high availability of MEC services is characterized by a high cost in terms of energy [60]. In order to minimize this cost, the SDN control plane programmability allows to make a tradeoff between energy cost and availability [61]. To comply with a high availability of MEC services such as in autonomous vehicular networks [62], the SDN controller can efficiently allocate computing, storage resources available in connected vehicles and MEC servers. The SDN control plane can also guarantee good availability by setting up efficient data routing paths, which contain as few network nodes as possible [63].

c: RESILIENCE

Resilience is the ability of the system to maintain its operational capabilities and to recover from hardware or software failures within an acceptable delay. MEC resilience relates to its fault tolerance and its capacity to continuously respond to multiple service requests without any disruption. SDN allows transparency of recoveries and deals quickly with failures. Logically centralized in the control plane, SDN controllers maintain a global view of the network, including the state of all nodes and links [64]. As a result, a controller can detect any network failure and dynamically redirect the traffic to avoid the failed nodes or links by installing new forwarding rules in the data plane components [65]. For instance, to enhance resilience in industrial IoT network like the smart grid, an SDN framework is designed in [66]. The controller schedules new traffic paths when failures are detected. Also, mitigating failures can be reached by fairly distribute the processing load among available MEC servers [67]–[70]. An SDN controller can balance the load in MEC environment to avoid any server overload that could make it defective. Data plane component redundancy is an appropriate solution to improve resilience in networks, but it is costly in terms of resources. In [71], an SDN/NFV-based MEC network algorithm is proposed to place VNFs in a distributed data center. Due to the SDN control capability, the algorithm can optimally place VNFs by reducing redundant data center capacity while still maintaining a high resilience. In a smart grid scenario, authors in [72] focus on how SDN can supply resilience using redundant communication links. If a wired link fails, the SDN controller activates a backup wireless link with low delay and update forwarding rules to achieve a seamless recovery.

d: INTEROPERABILITY

Interoperability is the property that allows a system to interfere and share data with multiple systems and products that are technologically different without any restrictions. While IoT devices from several vendors are increasingly used and different IoT applications are supported by the MEC environment, the need for technological independence is crucial. Using different protocols, heterogeneous IoTs devices generate data in various formats and models, which leading to a lack of interoperability [73], [74]. However, to support and provide the required MEC services, these generated data need to be analyzed and interpreted. In the light of these challenges, SDN technology, through its logically centralized control plane and the standardized southbound interface, offers a high interoperability level between IoT devices or end users and the MEC environment. SDN is integrated in several IoT architecture [75]–[78] to bring interoperability and, thus, decrease the complexity of providing services and communications capabilities. For instance, to deal with the interoperability issues caused by heterogenous IoT devices, a layered SDN architecture is designed in [77]. Based on this SDN architecture, especially the standardized southbound and northbound interfaces, many IoT devices could be utilized for various applications and services. Moreover, the utilization of SDN allows the heterogenous radio access networks (RANs) (e.g. WiFi, LTE) to readily exploit a shared computing, storage and networking resources offered by a MEC environment. In [78], a novel architecture is introduced to facilitate the interfacing between different wireless technologies using SDN concepts. An SDN controller is placed behind the gateways of each technology to manage either the connectivity between these gateways or the provisioning of MEC services.

e: EXTENSIBILITY

An extensible system is the one that supports adding new functionalities and features, so that its structure is little or not affected. In this work, we distinguish two types of extensibility in the MEC ecosystem, vertical extensibility and horizontal extensibility.

· Vertical Extensibility: Considering the huge number of new requested services by various end-devices, MEC architecture is said to be vertically extensible if it is designed to suitably include new services and applications. For this reason, SDN allows MEC administrators to develop and insert new facilities through northbound applications. The Adoption of the SDN architecture in the MEC ecosystem paves the way to develop innovative applications that interact programmatically and directly with control plane entities. Several types of applications can be built, such as security [79]–[81], virtualization [82], troubleshooting [83], configuration and management [50]. All these applications leverage the global view of the network to communicate the desired behavior of the MEC network to SDN controllers.

· Horizontal Extensibility: We define the horizontal extensibility in a MEC ecosystem as the connectivity between two MEC environments. Indeed, some cases require that a MEC ecosystem must communicate with another one to benefit, for example, from an available service in that ecosystem or to offload the processing tasks. Due to the SDN capabilities, connectivity between two MECs becomes flexible and agile, whether through the radio access network or directly between data planes. To share a part of radio, backhaul, core network, authors in [84] propose a generic software defined wireless networking architecture. Such architecture requires a programmable interconnection between the core control plane entities (e.g. mobility management, subscriber servers, packet gateways, serving gateways, etc.) that the SDN control plane can provide. The SDN controller is connected to each of these entities and they are programmable through an API.

C. SFC IN MEC

1) SFC DEFINITION

The desire for accelerating the transition to software and programmable networks has pushed network research and development groups to develop another technology known as Service Function Chaining (SFC) [11]. SFC helps operators and service providers to dynamically create network services and steer traffic flows between them. Although this sounds similar to the NFV definition and concepts, SFC addresses the problem of providing end-to-end services across a chain of service functions. In fact, SFC is an appropriate tool to suitably interconnect two or more service functions in a particular order in the network. Furthermore, it is possible to chain virtualized service functions (i.e. VNFs) and those embedded in physical network nodes. Accordingly, SFC ensures that physical network functions, which continue to exist in network service delivery architectures, are not excluded.

In an SFC network, traffic flows are classified and subsequently processed according to this classification. These treatments are applied in the order of the chain and there may be a reclassification in one of these treatments, leading to another chain. The Internet Engineering Task Force (IETF)

SFC working group has developed an SFC architecture for creating, modifying, and destroying ordered service chains that are applied to network traffic flows [85]. The main components of the SFC architecture include classifiers, Service Function Forwarders (SFFs), Service Functions (SFs) and SFC control plane. The SFC control plane communicates with all these components and it is responsible for multiple tasks such as constructing service function paths, selecting SFs for a requested SFC, providing information to classifiers on how to classify traffic flows, installing traffic forwarding rules in SFFs according to service function path. Classifier performs classifications by matching the traffic flows against the policy defined by the control plane for subsequent application of the required set of network SFs. Service function is responsible for specific treatment and can be realized as a virtual element or be implemented in physical network elements. SFF forward traffic to the connected services functions according to service function path information or to the classifier when it is necessary.

2) SFC IN MEC

Currently, the rapid evolution of end user services makes the MEC application layer very dynamic. Moreover, the huge number of virtualization technology proposals in terms of architectures, implementations or deployments makes the use of SFC crucial in the MEC ecosystem [86]. SFC enables MEC to tailor a network service function to end user context with complete features to provide end-to-end services [87]. The integration of SFC into MEC is an appropriate approach to organize the deployment of service functions, realize the desired strategies, adapt applications as strategies or policies change, and rationally allocate resources to meet the required services. SFC enables a variety of MEC applications that can enhance the functionality of MEC in terms of resource optimization [88]–[90], security [91], [92] and availability [93], [94]. In order to satisfy users of a 5G network with MEC in terms of service delay, the work [88] addresses the inter-MEC handoff problem by making the right decision to place and migrate SFCs. When a user is changing his cell, the proposed algorithm decides which VNFs of SFCs should be migrated, to which MEC servers and how much resources should be allocated. These decisions are intended to minimize the interruption of services. To meet a high QoS for IoT applications, authors in [91] developed an approach to carefully place service functions of SFCs in the edge network. They propose a probabilistic logic programming approach that ranks the VNF chains locations based on the bandwidth, maximum end-to-end latency and security requirements. Taking advantage of the secured service chaining architecture, a new SFC architecture is introduced in [92] to satisfy various security requirements for mobile edge computing. This solution aims to provide mobile users with a real-time security service chaining. To obtain a good level of security and an optimized order of the security functions, a fuzzy inference system-based algorithm is used to properly order the security functions and thus create the security service chains. Zhu and Huang [94] tackle the MEC application availability problem by strategically placing VMs. The locations of the VMs hosting VNFs are chosen to minimize the cost of resources, but not at the detriment of service availability. When a MEC application is requested, the proposed algorithm selects the host or the location that meet both the availability constraint and the resource requirements. Dinh and Kim [93] study the same problem, i.e. application availability issue in MEC environment, with emphasis on VNFs failures. To overcome this issue, they opt for the redundant deployment of some VNFs instead of the whole SFC. The operation of selecting the VNFs to redundant is based on availability metrics, which depends on the mean time between failures and the mean time required to repair the failures. Considering the limited resources of the edge network and based on this metric, a VNF redundancy scheme is developed.

D. NETWORK SLICING IN MEC

1) NETWORK SLICING DEFINITION

Network slicing is a specific form of network virtualization that consolidates the transition from a static network infrastructure to a much more dynamic infrastructure. It provides the ability to build several independent logical networks, called network slices, that run on top of a common physical infrastructure [95]. Each network slice is tailored and dedicated to support a specific service with distinctive characteristics and requirements. International Telecommunication Union (ITU) has classified these service requirements into three categories: (i) Ultra-Reliable Low-Latency Communications (URLLC); (ii) enhanced Mobile Broadband (eMBB); and (iii) massive Machine Type Communications (mMTC). The first category represents mission-critical applications such as autonomous driving and remote surgery, which require sub-millisecond latency with error rates that are lower than 1 packet loss in 10^5 packets [96]. The second category concerns high data rate applications like virtual/augmented reality and large-scale video streaming. The third category focuses on providing connectivity to a large number of devices with sporadic traffic such as smart metering, sensing and monitoring applications, so latency and throughput are not a big concern for this type of applications. SDN and NFV serve as a basis for network slicing by providing the lifecycle management of network slices. Network slicing leverages the SDN/NFV orchestration framework to dynamically instantiate, modify and terminate network slices so that each slice is the combination of multiple chained VNFs build up to support the service that the slice delivers to its end-user.

3GPP has introduced, through several 5G technical specifications and studies [25], [97]–[101], the main concepts and bases to seamlessly integrate and manage network slicing in the 5G RAN and 5G core network. The main network slicing concepts defined by 3GPP include Network Slice (NS), Network Slice Instance (NSI), Network Slice Subnet (NSS) and Network Slice Subnet Instance (NSSI). A NS is a logical network that provides specific network capabilities to

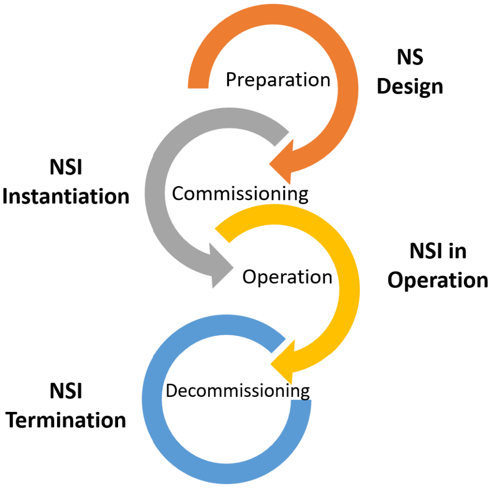


FIGURE 11. 3GPP NSI management lifecycle.

support the various characteristics of a service required by an end user. An NSI is a set of network function instances and the corresponding physical and logical resource to run these network functions instances (i.e. computing, storage and networking resources) which form a complete instantiated NS. An NSS can be defined as a sub-NS representing a logical network segment that it is part of a network slice. For instance, in an end-to-end mobile 5G network, a NS is composed of an access NSS and a core NSS. Similarly, a NSSI forms part of the overall network slice instance and one NSSI can be shared by several NSIs. 3GPP has introduced a four-phase framework to manage the lifecycle of NSIs. The four main phases, as depicted below in Fig. 11, are: (i) preparation; (ii) commissioning; (iii) operation; and (iv) decommissioning. The first phase represents a design stage of the desired NSI in which the network environment required to host this NSI is in preparation (i.e. the NSI is not yet created). In the second phase, the NSI is instantiated by allocating and configuring all needed resources to meet the NS requirements. The third phase includes four actions: (1) activation of NSI to be ready to support the service; (2) supervision and reporting actions consist in looking after the NSI KPIs; (3) modification action such as resource capacity or topology changes can be triggered as a result of supervision and reporting actions or to meet new received network slice requirements; (4) deactivation action can be performed to make the NSI inactive. The fourth phase can be activated when the NSI is no longer needed and it includes removing the NSI configuration and releasing the reserved resources.

3GPP Service and System Aspects group 5 (SA WG5) has described how NSIs and NSSIs are automatically managed by defining three management functions, namely communication service management function (CSMF), network slice management function (NSMF) and a network slice subnet management function (NSSMF).

- CSMF translates the requirements of a communication service to network slice related requirements (i.e. network type, network capacity, QoS requirements, etc.) for all deployment scenarios. It communicates with NSMF to transmit and update these requirements as well as to receive notifications, from NSMF, about any changes of the capability to support the network slice related requirements.
- NSMF manages the lifecycle of NSIs according to the NS requirements received from the CSMF and derives from these requirements the NSS related requirements.
- NSSMF manages the NSSIs based on the NSS requirements from the NSMF.

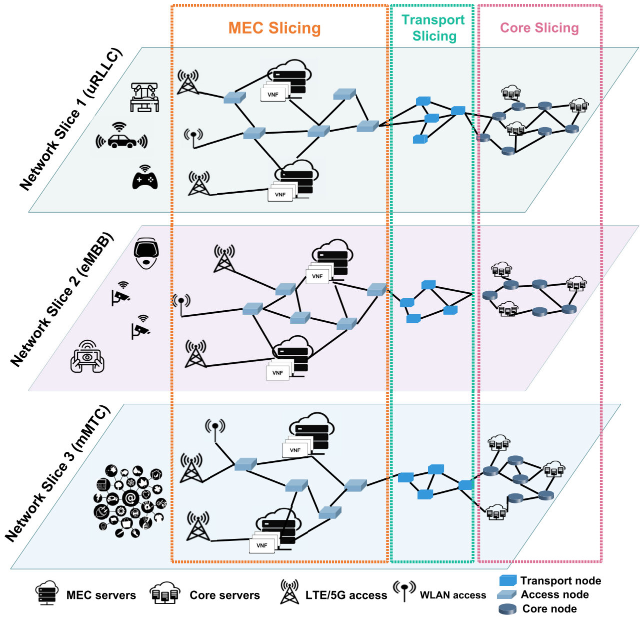


FIGURE 12. MEC network slicing in different 5G applications.

2) NETWORK SLICING IN MEC

While MEC enables end-user applications to benefit from cloud-based IT services at the edge of the network, it still needs an efficient configuration of these services according to the requirements of the end user. To overcome this challenge, network slicing can be used to dedicate optimal network resources to MEC applications. Indeed, with the network slicing ability, the MEC resources can be dynamically partitioned into different NSs, as shown in Fig. 12, that meet the stringent performance requirements (e.g. ultra-low latency and high bandwidth) of each particular MEC application. In order to describe how network slicing can be addressed in a MEC environment, ETSI MEC group has provided in [102] some important use cases and example of network slicing integration with MEC components. As a MEC enabler, network slicing provides two major benefits to the MEC environment: i) dynamicity and ii) efficient use of resources.

a: DYNAMICITY

Coupled with several standards of existing and emerging technologies, notably SDN and NFV, network slicing allows operators to dynamically design, deploy and customize various network slices on a common infrastructure to achieve a substantial variety of performance for different use cases. The dynamicity provided by the network slicing paradigm to the MEC environment is due to its process of managing the life cycle of network slice instances, as illustrated in Fig. 10, which allows operators to have full-service assurance capabilities. To benefit from this dynamicity enabled by network slicing, some architectural challenges need to be addressed by identifying new MEC functionalities or interfaces as well as changes to existing MEC functional elements, interfaces and requirements. In particular, ETSI and 3GPP should offer both hard and soft network slicing capabilities to facilitate the integration of MEC and network slicing, which allows to support different service level agreements.

b: EFFICIENT USE OF RESOURCES

The major challenge that distinguishes MEC resource utilization from traditional resource allocation issues is that edge nodes have tightly-nested and strictly-constrained networking, computation and storage resources. To overcome this challenge, network slicing pave the way to an efficient utilization of MEC resources. Indeed, based on the network slicing paradigm, the operators can allocate the right amount of resources according to the network slice requirements. For instance, one network slice can be designed to deliver low latency and low data rate while the other network slice can be configured to deliver high throughput. Moreover, in the MEC-5G era, service providers are looking for leverage the influence of modern technologies to improve productivity and, at the same time, reduce expenses (OPEX) and investments (CAPEX). For this purpose, network service provides tailored services in a MEC environment to a wide variety of users, machines, industries, and verticals.

Important efforts [103]–[108] have recently been devoted to the introduction of network slicing in MEC. Authors of [104] proposed in a MEC orchestration/management architecture enabled 5G network slicing. In the proposed architecture, the MEC is considered as a sub-slice like the RAN, core and transport sub-slices which constitute an end-to-end NS. Furthermore, they highlighted and provided solutions related to the security and isolation issues of MEC-slicing. In [105], the authors have provided two solutions to enhance the slice-awareness of some MEC components. The first solution represents a signaling framework that supports interactions between MEC-NFV and network slicing entities. The second solution enables an inter-slice channel which facilitates the exchange of data between users belonging to two different NSIs. Inter-slice communication is required in certain cases, such as time-constrained safety alert messages in a V2X communication system. Authors of [109] proposed a framework for network slicing in MEC-oriented architectures, from the perspective of slice request admission and the revenue model in order to solve the operator's revenue increasing problem under traffic variations constraints. Authors proposed a Lyapunov based algorithm with no knowledge on the distribution of the traffic. The dynamic slice request admission decision is developed using heuristic approach and the power allocation is achieved through successive convex approximation. However, the work could be enhanced by considering even complex scenarios in which multiple orchestrators are involved.

Many works have proposed frameworks for network slicing for 5G services [106], [107], [110]–[112]. For instance, the work of [111] presented an end-to-end network slicing framework for 5G IoT networks in a MEC ecosystem. The framework allows flexibility and dynamicity for placement of services as VNFs. The work in [106] has proposed three architectural variants to integrate MEC into 5G network slicing environment. The first variant architecture consists in devoting to each NS an individual MEC platform manager. In the second variant architecture, a single MEC platform manager is responsible for handling all NSs. The third variant is a distributed management and orchestration architecture where all VNFs of a NS have their own managers which are connected to a global slice manager. In [107], a MEC slicing framework is proposed to instantiate slice services on edge nodes. The framework allows tenants, such as mobile and virtual network operators, to visualize the available MEC services and in which MEC hosts these services can be instantiated. Each tenant can request an amount of computing, storage or networking resources related to the desired slice, and the framework is responsible for admitting the slice request and instantiating the network slice. In [112], authors proposed a novel SliceNet framework, based on recent advances on network slicing to address challenges regarding the migration of eHealth telemedicine services to 5G networks. Authors highlighted the design and the prototype of a media-centric eHealth use case, considering a set of innovative enablers in order to achieve end-to-end QoS-aware network slicing. The work of [113], authors proposed an architecture for video CDN provisioning functionalities as a service over multiple cloud domains within a 5G context. Authors in [113] also introduced the concept of a CDN slice, which is considered as a CDN service instance created on content provider's request. This work could be extended to include resource scaling, optimized VNF placement algorithms, and efficient video adaptation schemes.

IV. MEC OPTIMIZATION APPROACHES

To achieve an efficient exploitation of MEC resources, several QoS parameters are considered in optimization approaches. These parameters differ according to the angle of view of each component that desires to benefit from the MEC environment, such as the service provider, the infrastructure provider and the end-user, etc. Accordingly, we analyze in this section the existing optimization approaches in the MEC environment based on the optimized QoS parameter. Then, we classify these approaches into latency optimization approaches, processing, memory and bandwidth optimization approaches, energy optimization approaches and combined optimization approaches. A summary of the different optimization approaches is provided in Table 2. Furthermore, Table 3 elaborates the paradigms considered by each work, namely SDN, NFV, SFC and network slicing, as MEC enablers and

TABLE 2. Summary of optimization approaches by types of optimized resources.

Optimization approach	Latency	Processing resource	Memory/storage resource	Bandwidth resource	Energy resource
Yala et al. [38]	•				
Cziva et al. [40]	•				
Yang et al. [41]	•	•		•	
Leivadeas et al. [43]	•	•	•		
Kaur et al. [61]	•			•	•
Peng et al. [62]		•	•	•	
Ford et al. [71]	•	•		•	
Chen et al. [88]	•				
Sun et al. [90]	•				
Li et al. [92]		•			
Zhu et al. [94]		•	•	•	
Dinh et al. [93]		•	•		
Nam et al. [114]	•				
Ren et al. [115]	•				
Yang et al. [116]	•				
Son et al. [44]	•				
Alameddine [117]	•				
Chen et al. [118]					•
Li et al. [119]		•	•	•	
Chen et al. [120]		•	•	•	
Ahn et al. [121]					•
Li et al. [122]					•
Liang et al. [123]					•
Yang et al. [124]					•
Liu et al. [125]	•	•		•	
Blanco et al [126]					•
Zhang et al. [127]					•
Chen et al. [128]	•				
Yang et al. [129]					•
Xu et al. [130]	•	•	•	•	
Guo et al. [131]	•				•
X. Tran et al. [132]	•	•			•
Zhang et al. [133]	•				•
Al-Shuwaili et al. [134]	•	•		•	•
Zhao et al. [135]		•			•
Liu et al. [136]	•				•
Zhang et al. [137]	•	•			•
Zhou et al. [138]		•	•	•	•
Zhang et al. [139]	•				•
Sardellitti et al. [140]					•
Alameddine et al. [117]	•	•			
Peng et al. [141]		•	•	•	
Wang et al. [142]	•	•		•	
Pan et al. [143]	•	•		•	•
Taleb et al. [113]	•	•			
Tun et al. [144]					•

it indicates which approach takes into account the characteristics and requirements of 5G. Table 4 summarizes the assumptions considered in each approach to formulate the optimization model.

A. OPTIMIZATION APPROACHES FOR LATENCY

In order to minimize the latency and maximize the availability for uRLLC services, Yala *et al.* [38] propose a VNF placement algorithm. The latency of a service is defined as the access latency of the physical machines that host the VMs composing this service. The availability of a service depends on the failure probability of physical and virtual machines. The problem is formulated as a multi-objective optimization considering a budget constraint (CPU resource and energy consumption), which limits the cost of deploying services. For solving this problem, a generic algorithm is adapted to make a tradeoff between low latency and high availability of services. The work of [40] investigates the allocation problem of the VNFs running on multiple hosts in order to minimize the expected total latency between end users and their respective VNFs. The proposed model dynamically places VNFs according to latency fluctuations, user demands and user mobility to meet the low end-to-end latency required between users and VNFs. This model also schedules VNF migration operations to place them in optimal locations. Looking for these optimal locations avoids unnecessary VNF migration operations and reduces the latency violation. To achieve the service delay required by users in a MEC-based 5G network, Chen and Liao [88] address the inter-MEC handover problem by appropriately choosing where to initially place VNFs,

TABLE 3. Summary of optimization approaches in the MEC environment by utilized paradigm.

Optimization approach	SDN	NFV	SFC	5G	Slicing
Yala et al. [38]		•		•	
Cziva et al. [40]		•		•	
Yang et al. [41]	•	•			
Leivadeas et al. [43]		•			
Kaur et al. [61]	•			•	
Peng et al. [62]	•	•			•
Ford et al. [71]	•	•		•	
Chen et al. [88]		•	•	•	
Sun et al. [90]	•	•			
Li et al. [92]	•		•		
Zhu et al. [94]		•	•	•	
Dinh et al. [93]	•	•	•		
Nam et al. [114]		•	•		
Ren et al. [115]				•	
Yang et al. [116]	•	•			
Son et al. [44]	•	•	•		
Alameddine et al. [117]	•			•	
Chen et al. [118]				•	
Li et al. [119]		•	•	•	
Chen et al. [120]		•			
Ahn et al. [121]		•		•	
Li et al. [122]	•				•
Liang et al. [123]	•				
Blanco et al. [126]		•		•	
Yang et al. [124]				•	
Liu et al. [125]				•	
Zhang et al. [127]				•	
Chen et al. [128]	•			•	
Guo et al. [131]				•	
Tran et al. [132]				•	
Zhang et al. [133]				•	
Zhao et al. [135]				•	
Wang et al. [143]	•	•		•	
Sanchez et al. [111]					•
Tun et al. [145]					•
Martiradonna et al. [146]					•
Taleb et al. [113]		•		•	•
Mena et al. [103]					•
Wang t al. [112]					•
Xiang et al. [108]				•	•
Tun et al. [144]					•

when VNFs should be migrated, where migration operations should be performed (i.e. destination MEC servers) and how many resources should be allocated. The problem is formulated as an optimization problem minimizing the cost of the handover, which is defined as the weighted sum of the service migration time and the service downtime and was solved using a heuristic algorithm. The heuristic algorithm consists of two components, an SFC placement algorithm and an SFC migration algorithm. Both algorithms place each SFC in MEC servers respecting the required chaining order and the service delay constraint. The work of [90] introduces a novel VNF mapping concept called workflow-like service request (WFR) in a an NFV-based edge computing environment in order to provide network services. Unlike SFC, a VNF in a WFR can have several input functions and several output functions. Furthermore, non-interdependent VNFs can be processed in parallel. The proposed workflow model is used to serve different requests with low latency. To build a WFR in an edge computing network, a Dynamic Minimum Response Time considering Same Level (DMRT_SL) algorithm has been proposed to place and map VNFs. DMRT_SL reduces the total latency of a created WFR by deploying VNFs that are in the same layer on the same substrate node.

Authors in [114] propose a clustered NFV service chaining scheme in order to optimize end-to-end service time in a MEC network. A stochastic model is used to cluster VNFs according to their popularity, which depends on the probability request of each data flow. In addition, the work proposes an optimization problem of the service chaining time to obtain the optimal number of MEC clusters. With the use of this clustering scheme, the traffic that needs to be routed through a service chain will be processed in the same cluster. As a result, the number of clusters and the communication delay can be reduced. For data analysis applications such as video compression, the work [115] improves the QoE for users by minimizing the delay required for compression and transmission data to store them in the edge. Three models, namely local compression, edge cloud compression and partial compression offloading, are presented. In the local compression model, the data is compressed locally within the end device and then transmitted to the edge for storage. In the edge cloud compression model, the data is directly transmitted to the edge and it will be compressed in the edge by optimally allocating computation resources. In the partial compression offloading model, part of the data is locally compressed, and the rest is compressed into the edge. The end-to-end delay of these three models varies according to the compression delay and the transmission delay. The partial compression model can be useful when the local compression delay is important due to the limited speed of the CPU device and when the transmission delay is significant because of the bandwidth capacity of the channel. For the purpose of supporting workload changes that affect service-level response time requirements of low-latency MEC applications, a capacity violation detection mechanism was used in [116] to find out when the service-hosting nodes reach their limits. This mechanism forecasts the future workload of the system and detects the service-hosting node whose capacity will be violated. Accordingly, two scenarios are intended: (1) if there are service-hosting nodes that can accommodate the future workload in terms of resource capacities and latency, the flows will be redirected to these nodes. (2) if the current system cannot support the future workload, new service-hosting nodes placement should be found to avoid any latency violation. Therefore, an online adaptative greedy heuristic algorithm is developed to simultaneously determine the new locations and balance the load to the new service-hosting nodes. A resource management algorithm is developed by [44] to efficiently provide VNFs with the necessary resources in an edge-cloud environment. The algorithm uses edge computing resources to meet the requirements of latency-sensitive applications and cloud resources for less sensitive ones. When an overload is detected on a VNF in the edge, the algorithm duplicates this VNF at the same location if the available resources in the edge

TABLE 4. ***(Continued)*** **Summary of assumptions considered by each optimization approach.**

Optimization approach	Assumptions
Yala et al. [38]	• Physical machines in the same data center have the same access latency. • Failure probability of a VM is independent of the other VMs and physical machines. • Failure probability of a physical machine is independent of the other physical machines. • Failure probabilities of VMs and physical machines are known by the operator. • The operating cost of a physical machine is fixed and does not depend on the workload or the number of hosted virtual machines.
Cziva et al. [40]	• VNFs can be placed to any host in the network. • Migration cost of a VNF is time independent. • Each VNF tolerates a latency violation that should not exceed a threshold.
Yang et al. [41]	• All NFV-enabled commodity servers are identical. • The workload of the MEC servers is perfectly predicted. • The access delay to a MEC service is presented as a function of network hops. • The resources available on the network can satisfy requests from all access points.
Leivadeas et al. [43]	• End-user devices randomly enter and exit from a service chain according to an identical and independent Poisson process. • Only three topologies that a service chain can follow. • The total processing capacity of a server can be fully utilized.
Kaur et al. [61]	• The energy consumption considered represents the energy consumed by the active switches only.
Peng et al. [62]	• Processing results of a migrated task will either be sent directly to the connected vehicle if it enters the service area of the new MEC server, or they will be sent back to the original MEC server to respond to the request.
Ford et al. [79]	• The inter-cell handover volume that occurs on the network is known. • To improve network resilience, only failures of data center nodes are considered. • If a data center becomes unavailable, additional capacity must be reserved on one or more secondary data centers.
Chen et al. [88]	• The system operates in a slotted structure and its timeline is discretized into time frames.
Li et al. [92]	• A VM hosts only one security function of a security service chain. • A VM can host multiple security functions of different security service chains.
Zhu et al. [94]	• The failure of a VM is independent of the failure of other VMs. • The failure of a host is independent of the failure of other hosts.
Dinh et al. [93]	• VNFs failures are independent because the availability / reliability of the service functions is independently configured. • Parallel dependency is used in VNF redundancy deployments. • Serial dependence is used in the construction of SFCs. • Redundant VNFs have the same availability as their corresponding primary VNFs.
Nam et al. [114]	• The MEC network traffic is self-similar with a data rate average and it is modeled based on the fractional Brownian motion. • The popularity of VNFs is periodically calculated. • For each VNF, the request pattern follows a Zipf distribution. • Each VNF in the cluster has only one copy.
Ren et al. [115]	• The channel gain and video size of all devices are known. • All devices and the edge cloud utilize the same video compression technology. • Channel gain is a random variable, independently and identically distributed. • The video segmenting, stitching, and storing delays are neglected since they are much shorter than both communication and computational delays. • Channel access follows the TDMA method. • Each device can only transmit on its own time-slot. • Transmission can only be performed when all video data is completely compressed. • The edge node can start compressing a video only if all the data has been completely received. • The local compression part or the edge compression part can be transmitted at any time, but not simultaneously.

TABLE 4. ***(Continued)*** **Summary of assumptions considered by each optimization approach.**

Yang et al. [116]	• Only stateless mobile multimedia applications are considered, so requests are forwarded, in case of handover, to the new service-hosting node without service migration. • All nodes have the same capacity in terms of CPU and memory. • The workload is predicted.
Son et al. [44]	• VNFs are stateless. • The network latency map is updated after each duplication operation.
Alameddine et al. [117]	• Each user equipment has only one task at a time. • Each IoT application can only handle one task of user equipment at a time. • The user equipment set does not change during the offloading period. • The serving eNodeB of a user equipment is the one that offers the best received signal quality. • The upload delay is predefined. • The download delay is neglected.
Li et al. [119]	• All VNF instances support multi-tenancy software architecture. • The network is considered initially unused. • Cloud data centers are always operating. • Cloud Data Center resources are considered unlimited.
Chen et al. [120]	• The topology of service chains has a linear form.
Ahn et al. [121]	• All VNFs are instantiated on the same virtualization infrastructure. • The popularity of VNFs follows Zipf's law. • Each VM can host only one VNF. • When a request arrives at a VNF, the VM that hosts it fully operates. • The MEC network traffic is self-similar with a data rate average and it is modeled based on the fractional Brownian motion.
Li et al. [122]	• Node mobility and handover are not considered. • A machine-type communication device has only one packet to send in each time slot.
Liang et al. [123]	• User mobility and handover are not considered. • A flow can reach to the destination through several wireless links. • The energy consumed by the MEC server is not considered.
Yang et al. [124]	• Mobility and handover of end-devices are not considered. • A user equipment sends its task to the MEC server through a femto relay base station.
Liu et al. [125]	• Each UE accesses the servers having higher channel gains than a threshold value. • Each offloaded task is processed by only one server, and at each server, a single UE's offloading tasks can only be computed by one CPU.
Blanco et al [126]	• Each network service has a mean aggregated traffic demand. • Each VM hosts only one VNF instance. • A VNF instance can be deployed on several VMs.
Zhang et al. [127] 112	• The energy consumption by the backhaul between the small cell base station and the macro cell base station is ignored. • The computing ability of a MEC server is constant for each offloading task.
Yang et al. [129]	• Interference between mobile devices is ignored. • A task cannot be split into subtasks. • The overhead of the output data is ignored. • The transmission power of a mobile device remains at a random level.
Xu et al. [130]	• Each service provider only manages one service. • All data center servers consume the same amount of resources for running a service.

TABLE 4. ***(Continued)*** **Summary of assumptions considered by each optimization approach.**

Guo et al. [131]	• The macro base station and the relay share the same frequency band. • During the computation offloading period, the mobile device remains covered by the base station and has only one computation task. • A task cannot be split into subtasks. • The transmission delay from the MEC server to the mobile device is neglected.
Tran et al. [132]	• Each user has only one computation task at a time. • A task cannot be split into subtasks. • The transmission delay from the MEC server to the mobile device is neglected.
Zhang et al. [133]	• The small cell base stations share the same frequency band. • Energy consumption and transmission delay from the MEC server to the smart mobile device are ignored.
Al-Shuwaili et al. [134]	• The offloaded applications share inputs, outputs and computational tasks, which depend on the tracker, Mapper and Object recognizer components. • The channel state information is assumed to remain constant for the frame duration. • The shared CPU cycles are assumed to be less than the minimum of CPU cycles running at the cloudlet. • The subset of bits to be transmitted from the cloudlet to the users is to be less than the output bits to be transmitted in multicast mode to all users.
Zhao et al. [135]	• The subchannels are homogeneous for each smart device. • The channel gains of different subchannels are the same for the same smart device, thus, different for different smart devices). • Equal power is allocated to each assigned subchannel. • We consider that there is a long period comprised of many time slots. The channel state and resource allocation are may change at every time slot. • Using the mean value of all time slots as the average state for the long period. • The energy consumption increases accordingly with the allocated CPU-cycle frequency, and that the energy consumption could be controlled by through the CPU-cycles frequency with DVS technique.
Liu et al. [136]	• The end-devices have some stochastic information about the channel conditions. • The edge device allocates a fixed and equal amount of CPU for each end-device. • End-device adopt a TDMA scheme for data transmission.
Zhou et al. [138]	• Each base station belongs to an independent infrastructure provider, and the licensed spectrum of each infrastructure provider is orthogonal. • Each infrastructure provider only needs to solve its own problem without exchange of channel state information. • The position of macro base station is fixed, and the positions of the small base stations are uniformly distributed within the covered area of the macro base station. • Each base station does not cache initially any content.
Zhang et al. [139]	• Delay sensitive tasks are considered to be completed at the present moment. • One mobile terminal can use one and only one subchannel in fractional frequency reuse based on Hungarian method and graph coloring method. • Mobile Terminals and base stations own only a single antenna. • The interference coming from macro base stations and small base stations in adjacent cells is considered to be constant.
Peng et al. [141]	• MEC server could be placed in different places (MBS, MeNB, Edge nodes or Core Network) for more adaptability of application environments. • The power transmission of the MeNB and the Wi-Fi AP are fixed, and full signal coverage is insured by the MeNB on the road segment under study. No overlapping between Wi-Fi APs. • The tasks are generated during different time slots and the vehicles distribution is exposed to changes with time. • Vehicles are only associated to an MeNB when there is no coverage of a Wi-Fi AP.
Wang et al. [143]	• The communication model between vehicle and MEC infrastructure is a frequency-flat block fading Rayleigh, with block length above the completion deadline of application. • Vehicles could acquire the offloading probability of each other from the previous stage of the game. • No consideration for the handover between MEC infrastructures. • An application is considered as a single task.

TABLE 4. *(Continued)* Summary of assumptions considered by each optimization approach.

Pan et al. [142]	• Each device is able to connect either to an edge server or alternatively to a remote cloud server for computation offloading. • The tasks could be executed either locally, at the edge server or offloaded to remote cloud infrastructures, to be processed in a parallel fashion. • The battery capacity of the mobile device is considered to be limited.
Taleb et al. [113]	• Content is assumed to be already stored in MEC servers. • The QoE monitoring is performed knowing that the users already have the appropriate functionality deployed in order to record the relevant input QoE parameters. • QoE is affected by the virtual resource flavors (two flavors with same resources, the cheapest will be chosen)

are sufficient. If there are not enough resources, the algorithm places the new VNF in the cloud. Once creating the new VNF either in the edge or in the cloud, the algorithm creates a network latency map from the source node to the VNF in the cloud and in the edge. After duplicating VNFs and designing the latency map, the VNF forwarder distribute the newly arrived packets according to the application latency conditions. To cope with tasks offloading, application resource allocation and task scheduling problems in a MEC environment, a dynamic task offloading, and scheduling approach is designed in [117] for IoT applications. This approach aims to maximize the number of accepted tasks to be handled by IoT applications hosted on MEC servers, while meeting latency requirements. Multiple delays are considered, namely upload delay, waiting delay in the buffer of the IoT application, processing delay and edge to edge delay. Edge-to-edge delay presents the time required to transmit an offloaded task from the serving eNodeB to another one if the serving eNodeB is not connected to a MEC server or if the connected MEC server to it is not able to process the task. To overcome these delays challenges, the proposed approach carefully selects the MEC servers on which each task should be offloaded, allocates computing resources to the IoT applications that will handle the tasks and schedules when each task should be processed on the shared IoT application. However, the work does not consider dynamic scenario where mobile UEs join and leave dynamically during an offloading period. Authors in [128] investigate task offloading in ultra-dense network with the objective of minimizing the delay while enlarging the battery of user's equipment. Task offloading problem is formulated as a mixed integer non-linear programming solved by transforming it into two subproblems (task placement and resource allocation).

Authors of [143] propose a non-cooperative game to make the vehicles able to determine their task offloading strategies in real-time within a dynamic vehicular ecosystem. They propose a design of the payoff function using the distance between the vehicle and MEC access point as a parameter to adjust the offloading probability. Moreover, they construct a distributed algorithm based on the computation offloading game model to maximize the utility of each vehicle. The proposed algorithm converges to a unique and stable equilibrium. However, this work could be extended to cover complex scenarios such as multiple vehicles and MEC infrastructure collaboration. From the perspective of vehicular communication, the work of [141] studies the allocation of the spectrum, computing, and storing resources jointly within a MEC-based vehicular network and leverage reinforcement learning to solve the formulated problem using the deep deterministic policy gradient and hierarchical learning architectures. Offline training is considered to learn the network dynamics and resource allocation decisions can be rapidly obtained to satisfy the quality-of-service (QoS) requirements. The authors of [142] tackle the offloading problem as a dependency-aware task offloading decision in MEC, with the aim of minimizing the execution time under energy consumption constraints. They propose Q-learning approach to adaptively learn to optimize the offloading decision scheme under energy consumption through interaction with the network ecosystem. However, the work of [142] focuses on proving the feasibility and effectiveness of the reinforcement-learning-based approach for MEC, and as the work of [143], does not consider more complex MEC use cases in which multi-users are considered.

B. OPTIMIZATION APPROACHES FOR PROCESSING, MEMORY AND BANDWIDTH RESOURCES

For a cooperative driving between connected and autonomous vehicles (CAVs), several information needs to be shared whether from on board sensors or V2V/V2I. According to CAVs applications, several requirements are imposed. For instance, low delay is required for safety-related applications and high throughput for infotainment applications. However, the computing and storage capabilities of CAVs are limited to handle a large amount of data and to ensure the QoS of various applications. To overcome these challenges, the authors of [62] propose an autonomous vehicular network (AVNET) architecture based on the coexistence of the SDN and NFV concepts in the MEC. By activating SDN control functions on MEC servers, computing, storage and radio resources can be utilized efficiently and sliced resources among the wireless access infrastructure can be deployed successfully. To further improve MEC resource utilization and to guarantee a heterogeneous QoS, a resource management scheme is introduced. The proposed scheme tries to maximize the utility of the network, which is defined as a tradeoff between reducing the migration cost of computing/storing tasks among MEC servers and increasing the computing/storing resource utilization. The level of security required by MEC network users can be achieved by splitting security services into multiple simple security functions based on the SFC concept. In [92] authors proposed a fuzzy interference system-based algorithm to find the appropriate deployment of the required security functions. This algorithm considers CPU utilization and processing delay as determining factors in choosing the best order of required security functions. Proper placement of VMs hosting VNFs reduces operating costs and increases the availability of MEC applications. Therefore, the authors in [94] formulate the VM location problem as a stochastic programming model minimizing the operating costs of MEC services. To solve this problem and for the proposed model to be applicable in real scenarios, they propose a heuristic algorithm. The proposed algorithm makes a tradeoff between the high availability of services and the low bandwidth cost of the network. The cost of network bandwidth is reduced by deploying the maximum number of virtual machines required for this application on the same host. Application high availability is ensured if the VMs required for this application are deployed on different hosts. Due to the limited resources in the edge network, the availability of MEC application is ensured in the work of [93] by the redundancy of some VNFs instead of the entire SFC. The deployment of primary VNFs is performed considering the cost efficiency and the predefined SFCs availability requirement. Then, the problem of deploying VNF redundancy is formulated as an optimization problem that minimizes the cost of resource allocation. To solve the problem of deploying primary VNFs and redundant VNFs, two algorithms have been presented.

In order to timely serve the demands of users from different locations and to reduce the total cost of the resource used, authors in [119] employ multi-tenancy technology to place and chain VNFs in the hierarchical and geo-distributed edge computing network. Multi-tenancy technology allows a VNF instance to host several VNF requests of the same type, which reduces the number of VNF instances to be created and, thus, the resource consumption on the network will be reduced. However, in the case of several SFCs, reducing the number of VNFs instances may result to longer routing paths between these VNFs, which increases the bandwidth consumption. Therefore, an optimization problem is formulated to find a compromise between bandwidth consumption and resource (CPU and memory resources) consumption. To further improve user satisfaction, the total latency is constrained. To solve this problem in polynomial time, authors propose a heuristic-based algorithm to place and chain VNFs. Authors in [120] propose a VNF placement scheme to avoid unnecessary wastes of computing, storage and bandwidth resources of edge servers. The main objective is to increase the rate of resource utilization by reducing fragmentation of resources on edge servers. The idea behind this proposal is that the remaining resources of each edge server should be as small as possible after the deployment of the VNFs. A heuristic algorithm is designed to find locations for new VNFs, so that after their deployment, the remaining capacity of the hosting servers is minimal.

C. OPTIMIZATION APPROACHES FOR ENERGY RESOURCES

To optimize the energy consumption of MEC servers, authors in [121] propose a power consumption clustering scheme and minimize the power consumption of MEC environment while maintaining the average processing time of the flows in the MEC servers under the required threshold. The power consumption of a MEC environment is estimated as the average CPU load of all its servers. The average CPU load of a MEC server is measured as the average number of received requests. For solving this optimization problem, the proposed scheme attempts to determine the optimal number of clusters to reduce the power consumption. To improve the energy efficiency of end-user devices in a MEC-based virtualized cellular network with machine to machine communications, a stochastic optimization scheme is presented in [122]. Based on Markov decision process, the proposed scheme aims to address the problem of random access with M2M communication by decreasing energy consumption and optimizing the allocation of computing resources to end-user devices. Therefore, each machine-type communication device can decide, depending on the energy consumed, whether or not to offload computing tasks to the MEC server. With the objective of optimizing energy consumption in a mobile edge computing and caching network, Liang *et al.* [123] formulate an optimization problem. Bandwidth provisioning and content source selection are jointly considered to improve the energy consumption of the network. When a node (i.e. base station and MEC server) is selected as the source node of a flow, the energy consumed by this node to support the flow includes the energy of operation and transmission energy. The operating energy consists of the computational power consumption and the fixed power consumption related to circuits, control signal, etc. The transmission energy depends on wireless and backhaul transmission powers. Several constraints are considered, so that the allocated resource of links cannot exceed the backhaul bandwidth and radio resource. The proposed problem is solved by dual-decomposition method and a convex optimization problem solver called ADMM.

To efficiently deploy a complete 5G service consisting of a collection of VNFs, an energy-aware VNF placement strategy is required. Authors in [126] design a CE-RAN environment with 5G small cells in which the energy consumption of the entire system to provide services is presented as the energy demand of all allocated VNFs. The power consumption of a VNF depends on the hosting VM's CPU utilization, the power consumption of the physical switches that forward the traffic and the consumed power by the small cell that contains the VNF. In addition, authors of [126] use a protection parameter called robust constraint to overestimate the allocated resources in order to anticipate the peaks of traffic. Authors in [124] investigate the problem of minimizing the energy cost associated with offloading tasks under the delay requirements and considering the resource consumption as a constraint. The work of [118] proposes the minimization of the overall consumed energy of mobile devices while the work in [132] considers an objective function that can be parameterized to optimize the overall energy consumption and the overall task execution time. The total energy cost of user equipment, femto-relay base stations (FRS) and macro base stations with MEC servers (MEC-MBS) include the energy of the computing task and the energy of the transmission task. The transmission energy cost depends on the power of UE transmission over its fronthaul link to FRS and the transmission power of FRS over it backhaul link to MEC-MBS. The computation energy cost depends on the energy cost of the MEC servers for the processing tasks. The latency of a user equipment is considered and depends on the transmission latency between UE and FRS, queuing delay in FRS, transmission latency between FRS and MEC-MBS and computing latency in the MEC servers. The same problem is investigated in [127] for 5G heterogenous networks to improve energy efficiency in the case of offloading computation tasks. Depending on the energy cost, each end device takes the decision to either offload its task to the MEC servers or compute it locally. The energy cost in this case depends on the transmission energy between the end device and the MEC server and the energy consumed for the compute task, either in the end device or in the MEC server. The introduced optimization problem minimizes the energy cost of the whole system while ensuring the latency constraints of computing tasks. To solve the problem, an energy-efficient scheme is designed which classifies the end devices into three types based on the latency and energy costs of the tasks. The first type of devices offloads their tasks to the MEC server due to the limitations in terms of computation resources. The second type computes their tasks locally as long as the energy and the latency expended locally are lower than the fixed thresholds for the energy and latency, respectively, in the case of offloading. The third type either offloads their tasks or processes it locally based on the wireless communication conditions. The work in in [129] investigates the transmission energy consumed by end-devices and formulates the optimization problem to meet the QoE of end-devices under the computation delay constraint of the task. To minimize the transmission energy consumption, an offload strategy is investigated to decide on which MEC server an end-device must offload its task. Two algorithms are proposed: (1) to obtain the global optimal solution, a bound improving branch-and bound (BnB) algorithm is used; (2) to avoid the limited convergence speed of the first algorithm, especially in large scale network, a fast heuristic greedy algorithm is proposed.

D. OPTIMIZATION APPROACHES FOR COMBINED RESOURCES

The work in [41] investigates low latency requirements of mobile services and MEC cost efficiency. It studies the placement, moment and the manner of MEC resource allocation. The problem is formulated as an integer linear program minimizing the number of active MEC servers. Multiple parameters such as the link capacity, the number of network hops that the request traverses and the aggregated resource demands from access points are constrained to avoid any resource violation (e.g. bandwidth and CPU). In [43] a VNF placement approach is proposed which considers the existence of edge servers and cloud servers in order to provide network services to IoT applications. The VNF placement problem is formulated as a mixed integer programming problem minimizing hardware deployment costs while preserving the required communication delay. The end-to-end communication delay is defined as the combination of propagation delay, transmission delay, processing delay and queuing delay. Furthermore, based on Tabu Search meta-heuristic, a suboptimal algorithm is developed to obtain faster VFN placement solutions. The proposed approach is evaluated in a realistic environment using real resource characteristics and IoT requirements.

Authors in [134] propose a resource allocation scheme for computational and communication resources in order to optimize the mobile energy consumption. They also propose a Successive Convex Approximation (SCA) solution for the problem of energy resource consumption and convex-based approximation implementation for their approach. Then, they compare the proposed approach to a conventional independent offloading across users. Unlike the work in [147] and [140] that considers the general purpose application, the work in [134] investigates the cases of augmented reality where the generated data is combined with physical reality entities through sharing portions of computational tasks, input data and output data [148] and [149]. Challenges related to energy consumption and the assurance of a good QoS resulting from the coexistence between edge and cloud environments are overcome with the help of SDN technology in [61]. The authors consider two QoS parameters (latency and bandwidth) by which the SDN control plane classifies the flows before routing them to edge or cloud data centers. This classification divides flows into two categories, batch processing and flow processing. The first category requires bandwidth as a quality of service parameter, whereas the second category is latency sensitive. Also, they present two adaptive control plane strategies to make a tradeoff between energy efficiency and latency and a tradeoff between energy efficiency and bandwidth, respectively, to minimize energy consumption while ensuring QoS in terms of latency and bandwidth. The problem is decomposed into multiple subproblems to be solved simultaneously using the Tchebycheff method. Based on this method, the SDN controller schedules how to route flows across the network. Authors in [71] propose an optimization approach to meet the latency and resilience requirements of MEC applications while minimizing the cost of service migration and resources in terms of processing capacity and bandwidth. Resilience is ensured by reserving bandwidth and processing capacity resources for provisioning secondary data centers and links if the primary ones fail. The handover rate of users between cells is also considered to respect the latency requirements of MEC applications. For solving this problem, a set of heuristic algorithms are presented to allocate resources and properly route users' traffic to DCs. A new model that dissociates the infrastructure management tasks handled by the edge computing infrastructure provider (ECIP) from service management performed by the service provider (SP) is proposed in [130]. The aim of this work is to meet low-latency computing of time-sensitive applications and effectively leverage micro data centers resources in an edge computing network. With this model, an auction-based resource sharing contract was developed to maximize the utility of the SP and ECIP. SP utility is modulated as the gain in changing the execution of the real-time service from the cloud to the edge. ECIP utility was expressed as the profit that it can obtain by renting the resource to the SPs. A latency-aware task scheduling mechanism is designed to allocate the resources defined in the contracts.

Authors in [135] consider a case of MEC system with multi-mobile-users, where multiple smart devices request a computation offloading from a MEC server. The work proposes a novel method to jointly optimize the offloading selection, resource allocation of the radio resources, and computational resource allocation. It formulates the problem of energy consumption as a mixed integer nonlinear programming (MINLP) under latency constraints. As a solution, authors propose a reformulation-linearization-technique based Branch-and-Bound (RLTBB) method, which guarantees at least a suboptimal solution. They also propose a Gini coefficient-based greedy heuristic (GCGH) to solve the MINLP problem in polynomial time. Authors in [125] investigate the task offloading by considering the tradeoff between power and delay taking into account latency and reliability constraints based on Lyapunov optimization tools. They use a probabilistic model on users' queue length and exploit the extreme value theory to study low probability events from the perspective of queue length violation. Authors in [131] define four offloading decisions for mobile devices (MD) in a MEC-based 5G heterogenous network in order to minimize the computation overhead in terms of the processing delay and the energy overhead. The four decisions include that the MD decides to: (i) execute locally its computation task, (ii) offload its task to the MEC server connected to the small base station, (iii) offload its task to the MEC server connected to the macro base station and (iv) offload its task indirectly to the MEC server connected to the macro base station through a relay and the wireless backhaul. This decision problem is formulated as an optimization problem reducing the computation overhead of all MDs and is solved using a theoretical-game-based approach. The players in the game are the mobile devices, where each player chooses an offloading decision (strategy) according to the decisions of the other players until achieving an equilibrium state. The offloading utility of each end-user in a multi-server MEC network is formulated in terms of task computation time and energy consumption by the end-user device [132]. The task's computation time and energy consumption depend on the decision made by the device, i.e. offload the task to the MEC server or process it locally. If a task is uploaded to a MEC server, the time and energy required to transmit it to a MEC server are included. To optimize the offloading utility, authors propose a mixed integer nonlinear problem program. Due to the exponential time required to resolve the problem, it is decomposed to a task offloading problem and a resource allocation problem. The letter is decomposed to a computing resource allocation problem and an uplink power allocation problem. Authors in [136] consider resource allocation in an IoT context. End-devices act as agents capable of making decision regarding the offload of processing to the edge. The parameters considered in this article are the power consumed by end-device, latency of task's execution, channel condition between end-devices and gateways, and task queue. The decision-making is modeled as an MDP process which is solved by a reinforcement learning approach. Also, the authors propose an epsilon-greedy Q-learning algorithm for tasks offloading. The work in [137] investigates a joint optimization problem to achieve an optimal resource allocation scheme in a distributed fashion for defining the adequate pricing scheme. The problem is formulated as a many-to-many matching game to solve the pairing problem between the DSOs and the FNs and between the FNs and DSS. The authors of [138] propose a novel information-centric heterogeneous networks approach to enable content caching and computing. The proposed approach aims to enable caching and computing in a virtualized fashion which allocation strategy is solved by a joint optimization problem considering caching and computing. The work in [139] proposes a distributed joint computation offloading and resource allocation optimization approach for MEC heterogeneous networks. The proposed approach aims to finding the optimal computation offloading policy, which means the allocation of uplink subchannel and the power of transmission, and scheduling computation resources through a cloud and wireless resource allocation algorithm. The work of [133] investigates the tradeoff between the energy consumption of smart mobile devices and the latency required by their task. The weighted sum of energy consumption and execution latency is adjusted using the battery residual energy rate of the smart mobile device, which considers the battery's real-time service condition. Two scenarios are considered, the single and multicell MEC networks. In the multicell MEC network, interference management and channel assignment are considered in the optimization problem, while in the single cell MEC network, mobile devices do not deal with interference.

V. PROPOSED SDN-BASED MEC-NFV ARCHITECTURAL FRAMEWORK

In order to present our proposed MEC-NFV architectural framework based on the SDN architecture, in this section, we start by presenting an overview of the current efforts of the ETSI organization to design a seamless framework that allows the coexistence between MEC applications and NFV virtualized network functions. Then, we provide a detailed description of where an SDN controller can be placed to boost performance and increase network programmability.

A. MEC REFERENCE ARCHITECTURE

Fig. 13 illustrates the main functional elements and reference points of the MEC architecture designed by the ETSI Industry Specification Group (ISG) [27]. The MEC host level contains the VIM, the MEC platform manager and the MEC host. The VIM manages the lifecycle, performance and failures of virtual resources. The MEC host includes MEC applications, the MEC platform and the virtualization infrastructure for instantiating MEC applications by providing the required computing, storage and network resources. The MEC platform is responsible for several tasks, such as providing the MEC applications with the functionalities they need to consume and provide MEC services and offering the data plane with the traffic rules received from the MEC applications or the MEC platform manager. The MEC Platform Manager (MEPM) manages the lifecycle, requirements and rules of the MEC applications and provides element management functions and traffic rules to the MEC platform. The MEC system level contains the MEC Orchestrator (MEO), the Operations Support System (OSS), the device application and the customer facing service portal. The MEC orchestrator is responsible for the orchestrates of resources and services by using the Mm4 and Mm3 interface to communicate with the VIM and the MEC Platform Manager, respectively. Hence, the MEC orchestrator maintains a global view of the MEC system in terms of services, applications and available resources, allowing it to instantiate, relocate and terminate MEC applications according to their requirements. The OSS includes collection of functionalities that a MEC service provider needs to communicate with MEC device applications and third-party customers. The device application interacts with the MEC system to request services through a user application lifecycle management proxy. The customer facing service portal allows to third-party clients to request MEC services from the MEC system.

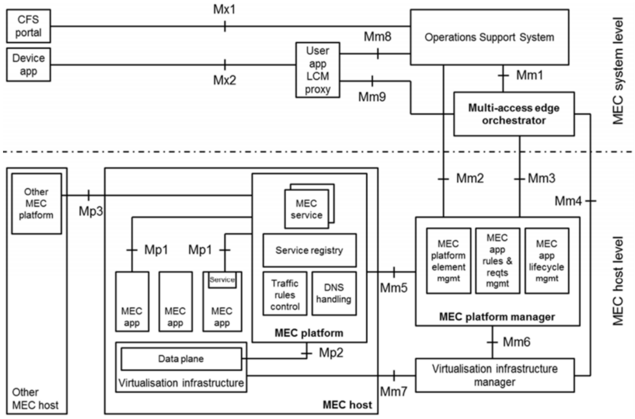


FIGURE 13. ETSI MEC reference architecture [27].

B. MEC REFERENCE ARCHITECTURE IN AN NFV ENVIRONMENT

As explained in Section III, MEC and NFV are intrinsically related and have mutual impact, which motivates the combination of their architectures. Accordingly, ETSI designed a consistent deployment of MEC in an NFV environment [27]. Fig. 14 depicts the deployment of the ETSI MEC reference architecture in the ETSI NFV environment. The proposed design makes the following changes in the MEC reference architecture: (i) the MEC applications are now considered as VNFs by the management and orchestration components (i.e. VIM, VNFM and NFVO); (ii) the MEC platform also became a VNF managed by a VNFM; (iii) the VIM manages the virtualization infrastructure that is deployed as an NFVI; (iv) since the MEC applications are now deployed as VNFs, the MEC platform manager assigns the MEC application (VNF) lifecycle to one or many VNFMs and is replaced by a new entity known as MEC platform manager-NFV; (v) the MEC Orchestrator is replaced by a new entity called MEC Application Orchestrator (MEAO) and is linked to the NFVO for the orchestration of resources and services. Furthermore, new reference points are thus deployed to ensure the communication between NFV components and MEC components such as Mv1, Mv2 and Mv3. We simplify, in Fig. 15, the ETSI MEC reference architecture in an NFV environment by preserving the blocks and reference points needed to design a MEC-NFV architectural framework based on the SDN architecture. The management entities, i.e. the VNF manager and the MEC platform manager, are

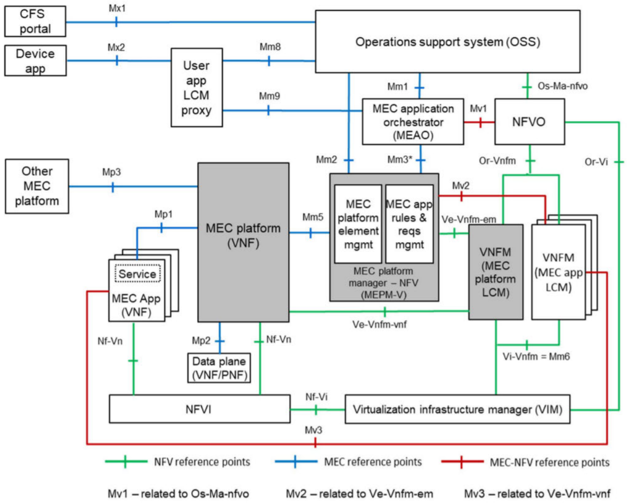


FIGURE 14. ETSI MEC reference architecture in an NFV environment [27].

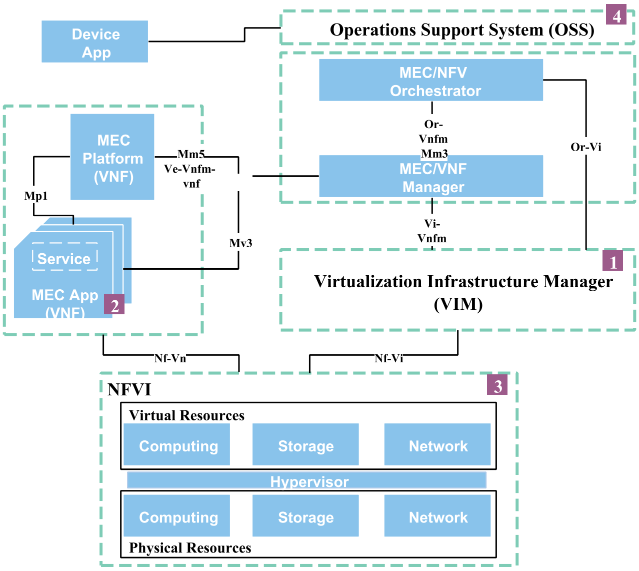


FIGURE 15. Simplified ETSI MEC reference architecture in an NFV environment.

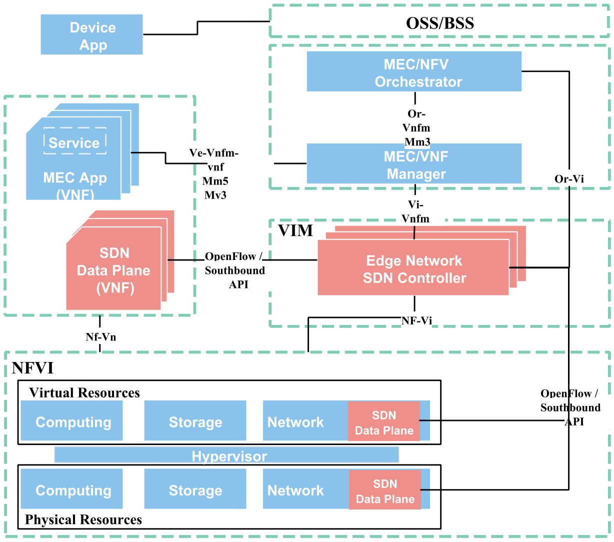


FIGURE 16. SDN-based MEC-NFV Architectural Framework.

grouped into a single block called MEC/VNF Manager. In the same way, the orchestration entities, i.e. the MEC application orchestrator and the NFV orchestrator, are presented by a single block called MEC/NFV Orchestrator.

C. PROPOSED SDN-BASED MEC-NFV ARCHITECTURAL FRAMEWORK

The SDN architecture defined by ONF [47] or International Telecommunication Union (ITU) [150] can be presented in 3 layers, namely the infrastructure layer, the control layer, and the application layer. For the interaction between these three layers, open Application Program Interfaces (APIs) are present. As previously stated in Section III, the infrastructure layer presents the data plane, which contains forwarding components (e.g. virtual or physical switches and routers) and it maintains connection with the control layer through an open southbound interface (e.g. OpenFlow protocol). The control layer presents the SDN controller that defines rules on how traffic should be forwarded and processed, and then pushes them down to components of the data plane. The top layer of the SDN architecture refers to applications that are directly consumable by end users. In this layer reside all programs that define the network behavior using the global network view provided by the SDN controller. In order to enhance the coexistence between NFV and SDN technologies, ETSI ISG have defined several proposals on how the NFV architectural framework, Fig. 10, can be designed on the basis of SDN architecture. Since the MEC architecture has been deployed in an NFV environment, the proposed designs for NFV framework continue to stand for the MEC-NFV architecture, Fig. 14. Multiple scenarios have been envisioned in [32] for the placement of the SDN controller in the MEC-NFV architecture. Four possible scenarios are illustrated in Fig. 15: (1) SDN controller is combined with the VIM functionalities; (2) SDN controller is deployed as VNF; (3) SDN controller is integrated into the NFVI; (4) SDN controller is integrated into the OSS. Merged with the VIM, the functionalities of the SDN controller will also cover those of the VIM. When deployed as a VNF, the SDN controller will be managed by a VNF manager (e.g. instantiation, update, query, scaling, termination). As part of NFV infrastructure (NFVI), the SDN controller will be responsible for network connectivity. The SDN controller can be part of the OSS and interface with the orchestration elements of the architecture.

Among the above-mentioned scenarios, we proposed in Fig. 16 an SDN-based MEC-NFV architectural framework. The SDN controller is placed with the virtualized infrastructure manager and the SDN data plane components can be either integrated in the NFV infrastructure or virtualized as VNFs. Choosing this placement of the SDN controller is in accordance with the ETSI recommendations. Indeed, ETSI recommends using existing interfaces (i.e. reference points) as much as possible to ensure communication between the different blocks of the architecture. Accordingly, placing the SDN controller in the VIM allows it to interact directly with three entities, namely the NFV infrastructure, the MEC/NFV manager and the MEC/NFV orchestrator, by exploiting available reference points. The SDN controller can use the Nf-Vi interface to directly manage the NFV infrastructure resources and perform operations on it. Resource management consists in configuring the infrastructure, providing virtualization enablers and scale up/down virtualized resources. Operations include collecting information about available resources and infrastructure failures to monitor the performance of the NFV infrastructure. The Vi-Vnfm interface can be used by the SDN controller to receive the resource allocation requests by the VNF managers that manages the lifecycle of the MEC applications and from VNF manager that manages the lifecycle of the MEC platform. The Or-Vi reference point connects the SDN controller with the NFV orchestrator. Therefore, the SDN controller can access to the repositories, namely network services catalog, VNF catalog, NFV instances and NFVI resources, which contain different information about resources, services and VNFs. Therefore, making the SDN controller as a part of the management and orchestration system facilitates its interaction with management and orchestration entities (e.g. VNF Managers and NFVO), which leads to introduce the desired programmability of the SDN controller. It is possible to have multiple SDN controller (e.g. a distributed SDN control plane) merged with the VIM. SDN Data plane components can be placed in different locations: (i) as physical components; (ii) as virtual components; (iii) virtualized as VNFs. For each location, the SDN controller can communicate with its data plane components either through a southbound API (e.g. protocol OpenFlow) or indirectly using existing MEC-NFV interfaces.

VI. CONCLUSION

The MEC paradigm is gaining momentum with telecommunication and IT ecosystems due to its ability to cope with latency and bandwidth issues and reduce the cost of transmitting data to the cloud. MEC provides an ecosystem of innovation for application developers and service providers and brings new levels of performance, especially for 5G network to support more IoT devices. To achieve its objectives, the functioning of the MEC network should be assisted by other technologies, in particular NFV, SDN, SFC and network slicing. In this article we covered the collaboration between 5G and MEC and the role of SDN, NFV, SFC and network slicing as complementary to MEC. This work summarizes the optimization approaches for MEC environment and exhibits a proposed MEC-NFV architectural framework based on the SDN architecture. Although several efforts have been spent to make MEC ready to meet all requirements, there still challenges such as standardization, efficient deployment, security and orchestration.

REFERENCES

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[4] 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," 2019, *arXiv:1906.08452*. [Online]. Available: http://arxiv.org/abs/1906.08452

[5] ETSI Group. *ETSI Multi-Access Edge Computing*. Accessed: Oct. 16, 2019. [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[6] Industrial Internet Consortium. *Fog And Edge Computing White Papers*. Accessed: Oct. 18, 2019. [Online]. Available: https://www.iiconsortium.org/fog-and-edge-white-papers.htm

[7] Open Edge Computing Initiative. *Open Edge Computing*. Accessed: Oct. 4, 2019. [Online]. Available: https://www.openedgecomputing.org/

[8] Open Networking Foundation. *CORD Project*. Accessed: Oct. 23, 2019. [Online]. Available: https://opennetworking.org/cord/

[9] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," 2014, *arXiv:1406.0440*. [Online]. Available: http://arxiv.org/abs/1406.0440

[10] B. Yi, X. Wang, K. Li, S. K. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.

[11] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, Nov. 2016.

[12] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[13] 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.

[14] 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.

[15] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[16] 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.

[17] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 260–288, 1st Quart., 2019.

[18] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: Open issues, challenges and future perspectives," *J. Cloud Comput.*, vol. 6, no. 1, p. 30, Dec. 2017.

[19] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017.

[20] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[21] L. M. Contreras, Y. Fang, W. Featherstone, D. Frydman, F. Jiangping, K. Kim, P. Kuure, A. Li, A. Odgers, D. Purkayastha, A. Ranjan, S. Scarpina, G. Verin, and K.-W. Wen, "MEC in 5G networks," p. 28.

[22] ETSI, "MEC deployments in 4G and evolution towards 5G," ETSI, Sophia Antipolis, France, White Paper, Feb. 2018.

[23] G. Cattaneo, F. Giust, C. Meani, D. Munaretto, and P. Paglierani, "Deploying CPU-intensive applications on MEC in NFV systems: The immersive video use case," *Computers*, vol. 7, no. 4, p. 55, Oct. 2018.

[24] Athonet, "SGW-LBO solution for MEC taking services to the edge," Athonet, White Paper, Feb. 2018.

[25] *5G; System Architecture for the 5G System (5GS)*, ETSI, Sophia Antipolis, France, Apr. 2019.

[26] *Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment*, ETSI, Sophia Antipolis, France, Feb. 2018.

[27] ETSI, "Multi-access edge computing (MEC); framework and reference architecture," ETSI, Sophia Antipolis, France, White Paper, Jan. 2019.

[28] *Developing Software for Multi-Access Edge Computing*, ETSI, Sophia Antipolis, France, Feb. 2019.

[29] *Mec in an Enterprise Setting: A Solution Outline*, , ETSI, Sophia Antipolis, France, Sep. 2018.

[30] ETSI, "Network functions virtualisation (NFV); architectural framework v1.1," ETSI, Sophia Antipolis, France, White Paper, Oct. 2013.

[31] ETSI, "Network functions virtualisation (NFV); architectural framework v1.2," ETSI, Sophia Antipolis, France, White Paper, Dec. 2014.

[32] ETSI, "Network functions virtualisation (NFV) release 3; evolution and ecosystem; report on network slicing support with ETSI NFV architecture framework," ETSI, Sophia Antipolis, France, White Paper, Dec. 2017.

[33] B. Blanco, J. O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino, E. Sfakianakis, F. Liberal, and G. Xilouris, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Comput. Standards Interface*, vol. 54, pp. 216–228, Nov. 2017.

[34] V. Sciancalepore, F. Giust, K. Samdanis, and Z. Yousaf, "A double-tier MEC-NFV architecture: Design and optimisation," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2016, pp. 1–6.

[35] G. Baldoni, P. Cruschelli, M. Paolino, C. C. Meixner, A. Albanese, A. Papageorgiou, H. Khalili, S. Siddiqui, and D. Simeonidou, "Edge computing enhancements in an NFV-based ecosystem for 5G neutral hosts," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–5.

[36] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[37] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping nfv-based multi-access edge computing in 5G ready networks with open baton," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–4.

[38] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and availability driven VNF placement in a MEC-NFV environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[39] I. Sarrigiannis, E. Kartsakli, K. Ramantas, A. Antonopoulos, and C. Verikoukis, "Application and network VNF migration in a MEC-enabled 5G architecture," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2018, pp. 1–6.

[40] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vNF placement at the network edge," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 693–701.

[41] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 475–488, Mar. 2018.

[42] R. Solozabal, B. Blanco, J. O. Fajardo, I. Taboada, F. Liberal, E. Jimeno, and J. G. Lloreda, "Design of virtual infrastructure manager with Novel VNF placement features for edge clouds in 5G," in *Engineering Applications of Neural Networks* (Communications in Computer and Information Science), G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds. Springer, 2017, pp. 669–679.

[43] A. Leivadeas, G. Kesidis, M. Ibnkahla, and I. Lambadaris, "VNF placement optimization at the edge and cloud," *Future Internet*, vol. 11, no. 3, p. 69, Mar. 2019.

[44] J. Son and R. Buyya, "Latency-aware virtualized network function provisioning for distributed edge clouds," *J. Syst. Softw.*, vol. 152, pp. 24–31, Jun. 2019.

[45] L. Ruiz, R. Durán, I. de Miguel, P. Khodashenas, J.-J. Pedreño-Manresa, N. Merayo, J. Aguado, P. Pavón-Marino, S. Siddiqui, J. Mata, P. Fernández, R. Lorenzo, and E. Abril, "A genetic algorithm for VNF provisioning in NFV-enabled Cloud/MEC RAN architectures," *Appl. Sci.*, vol. 8, no. 12, p. 2614, Dec. 2018.

[46] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Comput. Netw.*, vol. 81, pp. 79–95, Apr. 2015.

[47] G. P. Tank, A. Dixit, A. Vellanki, and D. Annapurna, "Software-defined networking-the new norm for networks," Tech. Rep., 2012.

[48] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.

[49] *Multi-Access Edge Computing (MEC); Framework and Reference Architecture*, ETSI, Sophia Antipolis, France, Jan. 2019.

[50] S. Peng, J. O. Fajardo, P. S. Khodashenas, B. Blanco, F. Liberal, C. Ruiz, C. Turyagyenda, M. Wilson, and S. Vadgama, "QoE-oriented mobile edge service management leveraging SDN and NFV," *Mobile Inf. Syst.*, vol. 2017, pp. 1–14, Jan. 2017.

[51] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low latency MEC framework for SDN-based LTE/LTE–A networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[52] E. Liotou, K. Samdanis, E. Pateromichelakis, N. Passas, and L. Merakos, "QoE-SDN APP: A rate-guided QoE-aware SDN-APP for HTTP adaptive video streaming," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 598–615, Mar. 2018.

[53] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.

[54] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Netw.*, vol. 32, no. 5, pp. 112–117, Sep. 2018.

[55] J. Al-Badarneh, Y. Jararweh, M. Al-Ayyoub, R. Fontes, M. Al-Smadi, and C. Rothenberg, "Cooperative mobile edge computing system for VANET-based software-defined content delivery," *Comput. Electr. Eng.*, vol. 71, pp. 388–397, Oct. 2018.

[56] C.-M. Huang, M.-S. Chiang, D.-T. Dao, W.-L. Su, S. Xu, and H. Zhou, "V2 V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17741–17755, 2018.

[57] A. Vladyko, A. Khakimov, A. Muthanna, A. A. Ateya, and A. Koucheryavy, "Distributed edge computing to assist ultra-low-latency VANET applications," *Future Internet*, vol. 11, no. 6, p. 128, Jun. 2019.

[58] A. Shaghaghi, M. Ali Kaafar, R. Buyya, and S. Jha, "Software-defined network (SDN) data plane security: Issues, solutions and future directions," 2018, *arXiv:1804.00262*. [Online]. Available: http://arxiv.org/abs/1804.00262

[59] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper, 2015, vol. 11, no. 11, pp. 1–16.

[60] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, "Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 48–56, Jun. 2017.

[61] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[62] H. Peng, Q. Ye, and X. S. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 156–162, Aug. 2019.

[63] A. Ateya, A. Muthanna, I. Gudkova, A. Abuarqoub, A. Vybornova, and A. Koucheryavy, "Development of intelligent core network for tactile Internet and future smart systems," *J. Sensor Actuator Netw.*, vol. 7, no. 1, p. 1, Jan. 2018.

[64] O. Blial, M. Ben Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," *J. Comput. Netw. Commun.*, vol. 2016, pp. 1–8, Apr. 2016.

[65] M. Kuźniar, P. Perešíni, and D. Kostić, "What you need to know about SDN flow tables," in *Passive and Active Measurement* (Lecture Notes in Computer Science). Springer, 2015, pp. 347–359.

[66] S. Al-Rubaye, E. Kadhum, Q. Ni, and A. Anpalagan, "Industrial Internet of Things driven by SDN platform for smart grid resiliency," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 267–277, Feb. 2019.

[67] L. Liu, S. Chan, G. Han, M. Guizani, and M. Bandai, "Performance modeling of representative load sharing schemes for clustered servers in multiaccess edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4880–4888, Jun. 2019.

[68] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[69] H. Wu, L. Chen, C. Shen, W. Wen, and J. Xu, "Online geographical load balancing for energy-harvesting mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[70] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 60–65, May 2018.

[71] R. Ford, A. Sridharan, R. Margolies, R. Jana, and S. Rangan, "Provisioning low latency, resilient mobile edge clouds for 5G," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2017, pp. 169–174.

[72] A. Aydeger, K. Akkaya, M. H. Cintuglu, A. S. Uluagac, and O. Mohammed, "Software defined networking for resilient communications in smart grid active distribution networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[73] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and open challenges," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 796–809, Jun. 2019.

[74] B. Ahlgren, M. Hidell, and E. C.-H. Ngai, "Internet of Things for smart cities: Interoperability and open data," *IEEE Internet Comput.*, vol. 20, no. 6, pp. 52–56, Nov. 2016.

[75] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 48–54, Sep. 2015.

[76] M. Ojo, D. Adami, and S. Giordano, "A SDN-IoT architecture with NFV implementation," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.

[77] Y. Li, X. Su, J. Riekki, T. Kanter, and R. Rahmani, "A SDN-based architecture for horizontal Internet of Things services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[78] M. B. Yassein, S. Aljawarneh, M. Al-Rousan, W. Mardini, and W. Al-Rashdan, "Combined software-defined network (SDN) and Internet of Things (IoT)," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 1–6.

[79] S. Shin, L. Xu, S. Hong, and G. Gu, "Enhancing network security through software defined networking (SDN)," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–9.

[80] L.-D. Chou, C.-W. Tseng, Y.-K. Huang, K.-C. Chen, T.-F. Ou, and C.-K. Yen, "A security service on-demand architecture in SDN," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2016, pp. 287–291.

[81] I. Farris, J. B. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin, "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 169–174.

[82] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna, and A. Moored, "5G-crosshaul: An SDN/NFV integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 38–45, Feb. 2017.

[83] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, "Fault-tolerance in the scope of software-defined networking (SDN)," *IEEE Access*, vol. 7, pp. 124474–124490, 2019.

[84] C. J. Bernardos, L. M. Contreras, H. Jin, and J. C. Zúñiga, "An architecture for software defined wireless networking," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 52–61, Jun. 2014.

[85] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," RFC, Tech. Rep., 2015, vol. 7665, pp. 1–32.

[86] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[87] L. Lei, X. Xiong, L. Hou, and K. Zheng, "Collaborative edge caching through service function chaining: Architecture and challenges," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 94–102, Jun. 2018.

[88] Y.-T. Chen and W. Liao, "Mobility-aware service function chaining in 5G wireless networks with mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[89] M. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Service chaining in MEC–Assisted large scale 5G networks," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2018, pp. 1–3.

[90] G. Sun, Y. Li, Y. Li, D. Liao, and V. Chang, "Low-latency orchestration for workflow-oriented service function chain in edge computing," *Future Gener. Comput. Syst.*, vol. 85, pp. 116–128, Aug. 2018.

[91] A. Brogi, S. Forti, and F. Paganelli, "Probabilistic QoS-aware placement of VNF chains at the edge," 2019, *arXiv:1906.00197*. [Online]. Available: http://arxiv.org/abs/1906.00197

[92] G. Li, H. Zhou, B. Feng, G. Li, T. Li, Q. Xu, and W. Quan, "Fuzzy theory based security service chaining for sustainable mobile-edge computing," *Mobile Inf. Syst.*, vol. 2017, 2017.

[93] N.-T. Dinh and Y. Kim, "An efficient availability guaranteed deployment scheme for IoT service chains over fog-core cloud networks," *Sensors*, vol. 18, no. 11, p. 3970, Nov. 2018.

[94] H. Zhu and C. Huang, "EdgePlace: Availability-aware placement for chained mobile edge applications: EdgePlace: Availability-aware placement for chained mobile edge applications," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 11, Art. no. e3504, Nov. 2018.

[95] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, 2020.

[96] *Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s)*, ETSI, Sophia Antipolis, France, Jan. 2019.

[97] *Management and Orchestration; Concepts, Use Cases and Requirements*, Standard TS 28.530 v16.2.0, Release 16, 3GPP, Jul. 2020.

[98] *System Architecture for the 5G System (5GS) Stage 2*, Standard TS 23.501 v16.5.0, Release 16, 3GPP, Jul. 2020.

[99] *Management and Orchestration; Provisioning*, Standard TS 28.531 v16.6.0, Release 16, 3GPP, Jul. 2020.

[100] *5G; Management and Orchestration; Provisioning*, ETSI, Sophia Antipolis, France, Oct. 2018.

[101] *Telecommunication Management; Study on Management and Orchestration of Network Slicing for Next Generation Network* Standard Specification # 28.801, 2016.

[102] *Multi-Access Edge Computing (MEC); Support for Network Slicing*, ETSI, Sophia Antipolis, France, Jan. 2019.

[103] M. P. Mena, A. Papageorgiou, L. Ochoa-Aday, S. Siddiqui, and G. Baldoni, "Enhancing the performance of 5G slicing operations via multi-tier orchestration," in *Proc. 23rd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2020, pp. 131–138.

[104] 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.

[105] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "MEC support for network slicing: Status and limitations from a standardization viewpoint," *IEEE Commun. Standards Mag.*, vol. 4, no. 2, pp. 22–30, Jun. 2020.

[106] L. Tomaszewski, S. Kukliński, and R. Kołakowski, "A new approach to 5G and MEC integration," in *Proc. Artif. Intell. Appl. Innov. AIAI IFIP WG Int. Workshops* in Advances in Information and Communication Technology, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham, Switzerland: Springer, 2020, pp. 15–24.

[107] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia, "Sl-EDGE: Network slicing at the edge," 2020, *arXiv:2005.00886*. [Online]. Available: http://arxiv.org/abs/2005.00886

[108] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5G networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[109] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du, and L. Zhu, "Dynamic network slicing and resource allocation in mobile edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7863–7878, Jul. 2020.

[110] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A State-of-the-Art review and framework," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 56–62, Mar. 2019.

[111] R. Sanchez-Iborra, S. Covaci, J. Santa, J. Sanchez-Gomez, J. Gallego-Madrid, and A. F. Skarmeta, "MEC-assisted end-to-end 5G-slicing for IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[112] Q. Wang *et al.*, "Enable advanced QoS-aware network slicing in 5G networks for slice-based media use cases," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 444–453, Jun. 2019.

[113] T. Taleb, P. A. Frangoudis, I. Benkacem, and A. Ksentini, "CDN slicing over a multi-domain edge cloud," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2010–2027, Sep. 2020.

[114] Y. Nam, S. Song, and J.-M. Chung, "Clustered NFV service chaining optimization in mobile edge clouds," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 350–353, Feb. 2017.

[115] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[116] B. Yang, W. K. Chai, G. Pavlou, and K. V. Katsaros, "Seamless support of low latency mobile applications with NFV-enabled mobile edge-cloud," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 136–141.

[117] 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.

[118] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.

[119] D. Li, P. Hong, K. Xue, and J. Pei, "Virtual network function placement and resource optimization in NFV and edge computing enabled networks," *Comput. Netw.*, vol. 152, pp. 12–24, Apr. 2019.

[120] Z. Chen, S. Zhang, C. Wang, Z. Qian, M. Xiao, J. Wu, and I. Jawhar, "A novel algorithm for NFV chain placement in edge computing environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[121] J. Ahn, J. Lee, S. Park, and H.-S. Park, "Power efficient clustering scheme for 5G mobile edge computing environment," *Mobile Netw. Appl.*, vol. 24, no. 2, pp. 643–652, Apr. 2019.

[122] M. Li, F. R. Yu, P. Si, H. Yao, E. Sun, and Y. Zhang, "Energy-efficient M2M communications with mobile edge computing in virtualized cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[123] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Energy-efficient resource allocation in software-defined mobile networks with mobile edge computing and caching," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2017, pp. 121–126.

[124] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[125] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–7.

[126] B. Blanco, I. Taboada, J. O. Fajardo, and F. Liberal, "A robust optimization based energy-aware virtual network function placement proposal for small cell 5G networks with mobile edge computing capabilities," *Mobile Inf. Syst.*, vol. 2017, 2017.

[127] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[128] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[129] X. Yang, X. Yu, H. Huang, and H. Zhu, "Energy efficiency based joint computation offloading and resource allocation in multi-access MEC systems," *IEEE Access*, vol. 7, pp. 117054–117062, 2019.

[130] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 47–54.

[131] H. Guo, J. Liu, and J. Zhang, "Efficient computation offloading for multi-access edge computing in 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[132] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[133] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[134] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[135] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[136] X. Liu, Z. Qin, and Y. Gao, "Resource allocation for edge computing in IoT networks via reinforcement learning," 2019, *arXiv:1903.01856*. [Online]. Available: http://arxiv.org/abs/1903.01856

[137] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[138] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.

[139] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[140] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[141] H. Peng and X. S. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, early access, Mar. 6, 2020, doi: 10.1109/TNSE.2020.2978856.

[142] S. Pan, Z. Zhang, Z. Zhang, and D. Zeng, "Dependency-aware computation offloading in mobile edge computing: A reinforcement learning approach," *IEEE Access*, vol. 7, pp. 134742–134753, 2019.

[143] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, and D. Zhao, "A game-based computation offloading method in vehicular multiaccess edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4987–4996, Jun. 2020.

[144] Y. K. Tun, M. Alsenwi, S. R. Pandey, C. W. Zaw, and C. S. Hong, "Energy efficient multi-tenant resource slicing in virtualized multi-access edge computing," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2019, pp. 1–4.

[145] Y. K. Tun, D. H. Kim, M. Alsenwi, N. H. Tran, Z. Han, and C. S. Hong, "Energy efficient communication and computation resource slicing for eMBB and URLLC coexistence in 5G and beyond," *IEEE Access*, vol. 8, pp. 136024–136035, 2020.

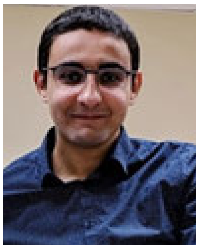
[146] S. Martiradonna, A. Abrardo, M. Moretti, G. Piro, and G. Boggia, "Architecting RAN slicing for URLLC: Design decisions and open issues," in *Proc. IEEE/ACM 23rd Int. Symp. Distrib. Simulation Real Time Appl. (DS-RT)*, Oct. 2019, pp. 1–4.

[147] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint Uplink/Downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 4, pp. 787–802, Dec. 2017.

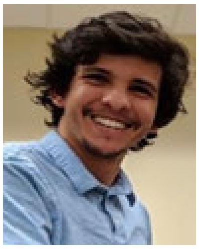
[148] D. W. F. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *Int. J. Virtual Reality*, vol. 9, no. 2, pp. 1–20, Jan. 2010.

[149] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 14–22, Jun. 2013.

[150] *Framework of Software-Defined Networking: Y.3300*, document ITU-T, Jul. 2014.



ABDERRAHIME FILALI received the B.Eng. degree in telecommunication and network engineering from the Ecole Nationale des Sciences Appliquée d'Oujda, Morocco, in 2016. He is currently pursuing the Ph.D. degree in computer engineering with the Department of Electrical and Computer Engineering, Université de Sherbrooke, Canada. His research interests include software-defined networks, network function virtualization, and heuristic algorithm design for next-generation networks. He has served as a Reviewer for several international conferences and journals.



AMINE ABOUAOMAR (Student Member, IEEE) received the B.Sc. degree in computer science and the M.Sc. of Research degree in computer science from Ibn Tofail University, Morocco, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Université de Sherbrooke, QC, Canada, and ENSIAS, Mohammed V University, Rabat, Morocco. His research interests include next-generation wireless networks and communications, edge computing networks, resource allocation, and networks virtualization.

SOUMAYA CHERKAOUI (Senior Member, IEEE) is currently a Full Professor with the Department of Electrical and Computer Engineering, Université de Sherbrooke, Canada, where she joined as a Faculty Member in 1999. She particularly works on next-generation networks (5G and beyond), edge computing/network intelligence, and communication networks for verticals, such as connected and autonomous cars, the IoT, and the Industrial IoT. Since 2005, she has been the Director of INTERLAB, a research group which conducts research funded both by government and industry. Before joining Université de Sherbrooke, she has worked for industry as the Project Leader on projects targeted at the Aerospace Industry. She avails of a long research experience in the wireless networking. Her work resulted in technology transfer to companies and to patented technology. She has published over 200 research papers in reputed journals and conferences. Her research and teaching interest includes wireless networks. She co-edited seven books and collective works. She is a Professional Engineer in Canada. Her work was awarded with recognitions and best paper awards, including the Best Paper Award at the IEEE Communications Society Flagship Conference IEEE ICC in 2017. She has chaired prestigious conferences such as IEEE LCN 2019, and has served as the Symposium Co-Chair for flagship conferences, including IEEE ICC 2021, IEEE ICC 2018, IEEE Globecom 2018, IEEE Globecom 2015, IEEE ICC 2014, and IEEE PIMRC 2011. She has been serving as the Chair for the IEEE Communications Society IoT-Ad hoc and Sensor Networks Technical Committee since 2020. She has been an Associate Editor and a Guest Editor of several IEEE, Wiley, and Elsevier journals. She is also on the Editorial Board of IEEE Systems, *IEEE Network Magazine*, IEEE Journal on Selected Areas in Communications (JSAC), and *Vehicular Communication* (Springer). She was named as a Distinguished Lecturer by the IEEE Communication Society.

ABDELLATIF KOBBANE (Senior Member, IEEE) received the M.S. (Research) degree in computer science, telecommunication, and multimedia from Mohammed V-Agdal University, Morocco, in 2003, and the Ph.D. degree in computer science from Mohammed V-Agdal University, and the University of Avignon, France, in September 2008. He has been a Full Professor with the Ecole Nationale Suprieure d'Informatique et d'Analyse des Systemes (ENSIAS), Mohammed V University, Rabat, Morocco, since 2009. He is also an Adjunct Professor with the L2TI Laboratory, Paris 13 University, France. His research interests include wireless networking, performance evaluation using advanced technique in game theory, and MDP in wireless mobile networks: the IoT, SDN and NFV, 5G networks, resources management in wireless mobile networks, cognitive radio, mobile computing, mobile social networks, caching and backhaul problem, beyond 5G, and future networks. He has more than ten years of computer sciences and Telecom experience, in Europe (France) and in Morocco, in the areas of performances evaluation in wireless mobile networks, mobile cloud networking, cognitive radio, ad-hoc networks, and future network 5G. He is the author of several scientific publications in top IEEE conferences and journals such as IEEE ICC, IEEE Globecom, IWCMC, ICNC, and IEEE WCNC. He is a Senior Member of ComSoc IEEE, an Ex-Secretary of ExCom IEEE Morocco Section, the Vice Chair of IEEE Communication Software Technical Committee, and the Ex-President and the Founder of Association of Research in Mobile Wireless networks and embedded systems (MobiTic) in Morocco. He is also the TPC Co-Chair of IEEE ICC 2020, the TPC Chair of Wireless Networking Symposium of the International Wireless Communications and Mobile Computing Conference (IWCMC 2019), the General Co-Chair of WINCOM 2020 and 2015, and the Executive Chair of WINCOM 2017.

MOHSEN GUIZANI (Fellow, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Computer Science and Engineering Department, Qatar University, Qatar. He has served in different academic and administrative positions at the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri-Kansas City, the University of Colorado-Boulder, and Syracuse University. He is the author of nine books and more than 600 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is a Senior Member of ACM. He also served as a member, the Chair, and the General Chair for a number of international conferences. He guest edited a number of special issues in IEEE journals and magazines. Throughout his career, he received three teaching awards and four research awards. He was a recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for outstanding contributions to the technological advancement of security. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He is currently the Editor-in-Chief of *IEEE Network Magazine*, serves on the editorial boards of several international technical journals, and the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* journal (Wiley). He served as the IEEE Computer Society Distinguished Speaker. He is currently the IEEE ComSoc Distinguished Lecturer.

• • •