IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 6, 2022, accepted February 11, 2022, date of publication February 18, 2022, date of current version March 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152787

A Survey on Mobile Edge Computing Infrastructure: Design, Resource Management, and Optimization Approaches

LINA A. HAIBEH, MUSTAPHA C. E. YAGOUB, (Senior Member, IEEE), AND ABDALLAH JARRAY, (Member, IEEE)
School of Electrical and Computer Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada

Corresponding author: Lina A. Haibeh (labou068@uottawa.ca)

This work was supported by the University of Ottawa.

ABSTRACT Emerging 5G cellular networks are expected to face a dramatic increase in the volume of mobile traffic and IoT user requests due to the massive growth in mobile devices and the emergence of new compute-intensive applications. Running high-intensive compute applications on resource-constrained mobile devices has recently become a major concern, given the constraints of finite computation and limited storage capacities. Mobile Edge Computing (MEC) has recently become the key technology to overcome these issues by providing cloud computing capabilities and placing IT infrastructures at the mobile network edge. In this survey, we present a list of relevant research papers for the MEC infrastructure implementation phases, including (1) MEC infrastructure designing and dimensioning, (2) MEC infrastructure virtualization using Network Function Virtualization (NFV) concept, and the use of virtualized service placement and auto-scaling methods to deploy an agile system framework, (3) MEC resource management frameworks, and (4) approaches used to optimize the MEC resources on the physical infrastructure. The main focus of this survey is to determine the required aspects to implement an auto-scaled and proactive MEC-NFV infrastructure to support a dynamic and heterogenous mobile users' demand at mobile network operators.

INDEX TERMS Mobile edge computing, MEC infrastructure design and dimensioning, mobile network operator, optimization approaches, proactive, resource management, virtualization, NFV, VNF autoscaling, VNF placement.

I. INTRODUCTION

The 5G era has witnessed tremendous growth in mobile subscriptions and mobile data traffic. Based on Ericsson's latest forecast for 2020–2026, it is expected that global mobile subscriptions will grow from 7.9 billion to 8.8 billion, and international mobile data traffic will double [1]. The increased mobile network coverage drives this significant increase in mobile data traffic, the massive IoT (Internet of Things) device deployments, and mobile broadband subscriptions. Therefore, the demand for traditional cloud-based services in mobile networks increases dramatically, such as video streaming, social networking, and online retail services. Moreover, there is a high demand for new cloud services, such as mobile cloud games, remote control services for air and ground vehicles, and services that handle manufacturing processes [2].

To meet the increasing demand for computational requests in cellular networks, conventional cloud computing platforms (e.g., based on data centers) are continuously expanding their servers' capacities and thus, improving the Quality of Service (QoS) they provide. However, for various reasons, new computing platforms and architectures are needed to better scale with the explosion of the mobile service requirements. For mobile users to access traditional cloud services, mobile traffic must pass through multiple stages, including the mobile backhaul and possibly the mobile core operator, which creates additional communication latency that can exceed the latency requirements of critical services. In addition, intensive investments in computing and communication resources are required for conventional cloud computing platforms to enhance the quality of the services provided in cellular networks, which further increases the costs of access to cellular networks [3]. Finally, network knowledge and user context information can help provide localized services to the end-users and enhance their Quality-of-Experience (QoE).

The associate editor coordinating the review of this manuscript and approving it for publication was Fung Po Tso.

Mobile Edge Computing (MEC) is emerging as a promising technology to meet these needs [4], where it can provide the cloud computing capabilities by placing computational infrastructures at the network edge (i.e., radio access network ''RAN'') near mobile users to offer real-time information. Taking advantage of the close distance to the mobile end users, the mobile operators can improve QoE for their users and provide context-aware services with the help of RAN information [5] to reduce the network delay. Furthermore, edge data centers also can co-work with centralized cloud data centers for service orchestration. Such decentralization and collaboration in MEC are expected to transform infrastructures and applications significantly [6].

In edge computing, where and how to place the edge server is a vital issue. An appropriate MEC infrastructure implementation process is then required. To this aim, one needs to (1) maximize the used computation resources at each edge node, (2) provide efficient real-time services to the mobile users, and (3) guarantee the quality of experience for the mobile users. To do so, we identified three stages to implement MEC infrastructure.

First, assuming no existing MEC infrastructure is already deployed next to the Base Station (BS) at the mobile network operator. MEC infrastructure is therefore built based on the workload predictions for the mobile traffic received from the end-users (mobile users, connected vehicles, or IoT devices) at each BS location. For example, the massive mobile traffic received at specific BS locations can increase network congestion on the cloud and negatively impact network performance. Deploying localized MEC infrastructure in high-traffic BS areas can benefit conventional cloud providers, mobile core operators, and Internet service providers.

Second, the MEC infrastructure interacts with the mobile end users and defines how it manages MEC computational resources to satisfy their needs. MEC resource management is a multi-objective management process that could be described as the way of deciding where the computing MEC resource for each user should be performed, how much of each resource is needed, and what resources should be allocated, taking into consideration: (1) the unpredictability of user mobility behavior, and (2) the dynamic properties of the network. Generally, resource management depends on other hardware or software architectures (for example, the MEC resource orchestrator) integrated into the MEC infrastructure to ensure optimal node selection for edge computing.

Third, MEC infrastructure should be optimized to meet the performance demand of the mission-critical applications offered by MEC. It should efficiently utilize its computational resources and infrastructure. In this stage, identifying which QoS parameters to optimize for each MEC environment component is essential, including the service provider, the infrastructure provider, and the mobile end-user. To successfully implement MEC infrastructure at mobile operators, these three broad steps focus on the following four major phases:

Phase 1: Estimate the volume of mobile traffic received from mobile end-users at a specific BS location, design, and size new MEC infrastructure, or expand the existing one.

Phase 2: Determine Virtualized Network Service (VNF) auto-scaling and placement mechanisms used in MEC-NFV infrastructure.

Phase 3: Install MEC framework using additional hardware and software for MEC resource management and provide VNF scaling decision.

Phase 4: Optimize MEC infrastructure to achieve adequate QoS for end-user requests.

To the best of our knowledge, most current surveys focus on investigating and presenting the works related to one or more of these phases but not all four phases. Our survey is different from the existing surveys as it presents a critical evaluation of solutions that contribute to the practical end-to-end implementation of MEC infrastructure in the NFV environment, including all these four MEC deployment phases. In other words, this survey provides a framework that covers all steps to implement an agile, auto-scaled, and cost-efficient MEC infrastructure at the distributed edge nodes to support the mobile traffic heterogeneity and dynamicity. Accordingly, our main contributions are summarized as follows:

- Present and discuss existing MEC infrastructure design and dimensioning techniques.
- Present a structured classification of virtualized services placement and auto-scaling methods in MEC-NFV infrastructure based on the MEC use case.
- Review current framework solutions for MEC infrastructure and resource management, including resource orchestration and centralized control for the distribution of MEC servers.
- Present the optimization approaches to solve VNF placement problems from three different points of view: end-user, infrastructure owner, and service provider.
- Considering no existing MEC infrastructure at the BS location, propose the main steps to deploy the MEC-NFV framework that dynamically place and proactively scale the VNFs at the distributed edge nodes to ensure better network performance and avoid service disruption. The proposed MEC-NFV framework adapts well to the dynamic nature and the heterogeneity of IoT traffic.

The publications listed in this survey will focus on the following topics: (1) MEC infrastructure size and design, (2) MEC infrastructure virtualization using NFV technique, (3) VNF placement and auto-scaling in MEC-NFV environment, (4) MEC resource management frameworks, and (5) Optimization approaches in MEC-NFV infrastructure.

The remaining of this survey is organized as follows. Section II in this survey explains the main components of MEC architecture and how they interact to achieve the MEC system benefits, such as supporting real-time applications with low latency, reducing mobile energy consumption, reducing service cost, and supporting users' mobility.

Section III lists the MEC applications and the main motivations to use the MEC. In addition, this section presents the key benefits and challenges of the MEC systems. Section IV examines and compares current findings on the design and dimensioning of MEC infrastructure and discusses the importance of deploying MEC system on NFV infrastructure to achieve its benefits and work efficiently. Section V discusses MEC frameworks from different perspectives: Managing MEC server distribution using virtualization technology (i.e., Software Defined Network ''SDN''), resource orchestration among MEC multi-sites, building an agile MEC framework that handles time-varying traffic using different auto-scaling and placement approaches and optimizing MEC-NFV framework architecture considering different QoS parameters. Section VI presents the learned lessons and proposes a guideline with the main steps to deploy an end-to-end MEC-NFV proactive auto-scaled architectural framework. Finally, Section VII concludes the paper.

II. MEC OVERVIEW

MEC is an example of delivering cloud service capabilities at the edge of mobile networks, close to the end-user [7]. According to the European Telecommunications Standards Institute (ETSI), the MEC is characterized by low-end latency, local computing and storage resources, network awareness, and enhanced service quality provided by mobile operators [8]. MEC requires seamless integration of both mobile network operators and the service providers in architectural design and resource management [8]. MEC's development is further driven by several incentives for market transformation, including the need for the mobile operators to reduce the time-to-market of the new compute-intensive applications to increase their profit. Thus, the participation of different parties (i.e., network service providers, mobile network operators, and mobile end-users) is required to ensure a successful MEC deployment.

Referring to ETSI white paper [9], MEC is differentiated by the following:

1) On-premises: MEC can be deployed and run in an isolated environment where it has access only to its local computing resources and is completely separated from the rest of the network.

2) Proximity: MEC servers are typically placed in proximity of mobile users. This close distancing would allow mobile operators to collect and store real-time information from mobile users and process it for different purposes such as big data analytics and support location-aware services.

3) Low latency: MEC system can reduce the propagation and communication latencies and avoid the network congestions on the front haul and backhaul network links. Thus, this will make it possible for MEC to be a key enabler for latency-critical 5G applications while enhancing the content and service responsiveness time.

This section discusses the MEC architecture from a flexible resource management perspective, including communication and computing resources.

A. MEC ARCHITECTURE

MEC delivers mobile end-user services in the form of computing and storage resources. MEC resources are expected to be deployed on mobile networks that are locally close to end-users. Specifically, MEC resources can be used at both indoors and outdoors base stations (BS), Access Points (AP), as well as radio access networks (RANs) that connect user equipment (UEs) to the core network of MNO [10]. The exact deployment of MEC resources depends on physical constraints (such as power supply, available space, and deployment budget), performance requirements, and network operators' preferences [10]. Considering the different MEC deployment options, Fig.1 presents MEC components in three categories based on their functions and how they interact to achieve MEC system goals [11]–[16]. The first category is the MEC end-users, including mobile users, connected cars, and Internet of Things (IoT) devices. The second category is computing components that are used to handle end-user applications. Finally, the end-user and computing components are connected by communication components, the third category of MEC components.

1) COMMUNICATION UNITS

These units include wireless and/or wired communication networks:

Wireless networks are considered the primary way to provide connectivity for end-users in the MEC. End-users can be either connected to an unlicensed radio spectrum (i.e., WiFi) or directly connected to base stations and mobile access points that operate on licensed radio spectrum [17]. For example, 5G networks should provide enhanced support for bandwidth-intensive applications in MEC [18]. Additionally, 5G networks support Ultra-Reliable Low Latency Communication (URLLC), which is expected to achieve 1-millisecond latency and support applications that require strict latency requirements (e.g., process control and control services for drones) [19]. In addition to connecting end-users, wireless is considered an alternative to a wired backhaul, so wireless can also be used to communicate to MEC resources [20].

As for wired networks, the mobile backhaul network interconnects the various RAN components and the mobile core, establishes communication between the MEC components and the base stations or access points, and finally, the end-users. The physical infrastructure consists of cables (i.e., fiber) and various forwarding and routing devices. Because transmission time is relatively short in wired environments, packet processing time on routers and switches constitutes a significant part of the delay.

In communication networks, network functions allow monitoring and managing communication resources. Typical networking functions include resource provisioning, orchestration, performance analysis, error detection, and load balancing.

Network Functions Virtualization (NFV) is becoming increasingly popular in the telecommunications industry, as it

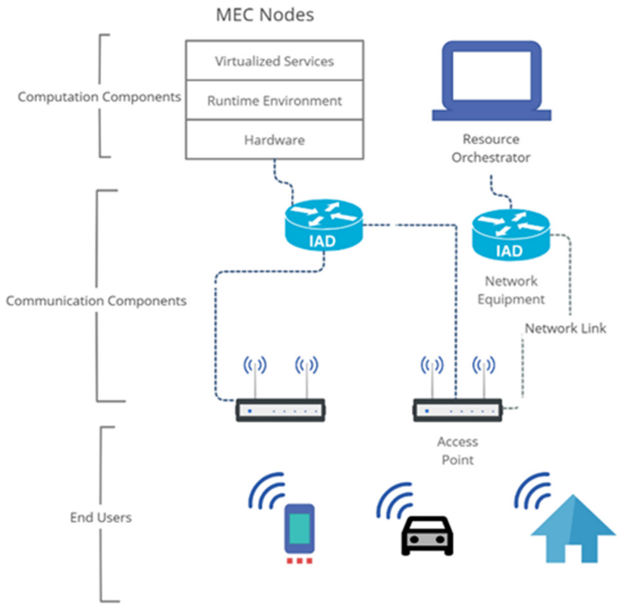

FIGURE 1. An illustration of main MEC components.

facilitates network management agility by virtualizing heterogeneous network functions provided by dedicated hardware [21]. Moreover, MEC-related network functions can also be virtualized [22].

2) COMPUTING COMPONENTS

These components are another MEC component category that provides computing and storage resources to serve MEC end users. These components include the MEC nodes and orchestrator:

a) MEC nodes: The MEC node may be heterogeneous depending on the performance requirements and deployment conditions. These nodes can have out-of-the-box commercial servers, small form factor servers, and purpose-built servers (for example, built-in artificial intelligence capability and high-volume storage capacity). MEC node consists of three components: Virtualized Services (VS) corresponding to the user application, a runtime environment that provides software support for running VS, and built-in hardware components.

Due to the tight integration of MEC and mobile networks, mobile network operators have the advantage of being owners and operators of MEC nodes. However, third-party service providers, such as property facility owners and cellphone tower owners, may have MEC nodes for their benefit, depending on the cost and complexity of deployment [23].

b) Resource orchestrators oversee and operate MEC nodes to efficiently use their computing resources and provide the expected QoS. The main resource orchestrator functions include resource allocation, virtual network service placement, task scheduling, and software updates. Additionally, resource orchestrators may be responsible for detecting failures in the MEC and enabling failed schemes [24].

III. MOTIVATION, APPLICATIONS, BENEFITS, AND CHALLENGES

This section lists the current computational application scenarios that force the use of MEC platforms besides the vital enablers that help MEC provide low-latency and context awareness services. Moreover, a list of key benefits and potential challenges for the MEC system are presented in this section. MEC platform is ideal for dynamic content optimization, computational offload in the IoT, extensive mobile data analysis, and intelligent transport. Considering that such applications are unsuitable for working on mobile or portable devices, they require large storage capacity and computational power.

A. MEC APPLICATIONS

Mobile edge computing has a high potential to offer a wide range of computing application types to its mobile users. The recent applications in MEC can be classified as offloading computing resources, collaborative computing, storage replication, and web content delivery [32]. These computing applications process the user's requests on the edge network, minimizing the network delay and enhancing service quality. The applications mentioned above use context information to enhance the user experience by offering heterogenous service types for mobile users.

1) COMPUTATIONAL OFFLOADING

Mobile applications like automatic speech recognition, video streaming, mobile games are considered compute-intensive applications. However, running this type of application on resource-limited devices requires a lot of computing resources and power. Instead, the task could be offloaded to the remote cloud the final result is returned when a task is completed successfully. Since the connection between the edge device and the remote cloud requires a long time, mobile edge computing servers could be deployed at the network edges with little use of resources. That way, high-intensive tasks can be offloaded.

Collaborative computing brings people and different companies and organizations together in a distributed computing system. In MEC, examples of collaborative computing applications range from simple sensors to robotically assisted and remotely controlled telesurgery. In such applications, device location and communication channel delay play a critical role during communication among different system users. Extending the real-time collaborative application in a mobile edge environment offers a robust context-sensitive collaboration model in the MEC system.

2) STORAGE REPLICATION

LTE has become the dominant technology for devices over the past year. IoT devices are less computational and have less storage capacity. These devices get the data over the network and offload it as a storage object for further computation in the scalable cloud infrastructure. As the number of IoT devices grows, the simultaneous replication of storage objects increases network latency. The edge servers can host multiple cloned clouds for each device, bringing computing power closer to the IoT device and reducing network delay.

3) CONTENT DELIVERY

It enhances web content on web servers to offer high availability, high quality of service, and reduce network delay. Conventional web content delivery cannot adapt to the end-user demands when optimized. MEC can support the dynamicity of web content optimization based on network conditions and the system load. The proximity of devices allows edge servers to utilize user mobility and QoE to optimize the web content.

B. KEY ENABLERS

The real-world implementation of mobile edge computing technology can be traced back to the support of various key technologies. The critical enabler attribute represents multiple technologies that help provide context-related services with low latency and high bandwidth for mobile network users near the RAN.

1) CLOUD AND VIRTUALIZATION

Virtualization creates an abstraction layer over the underlying hardware, introducing a logical infrastructure variant in the same physical hardware. The existence of computational resources at the network edge makes it possible to deploy different virtual machines using virtualization technology to offer different cloud computing services.

2) HIGH VOLUME EDGE SERVERS

In the MEC platform, powerful physical servers are collocated in each mobile base station at the edge network. These edge servers are responsible for processing the offloaded computational tasks from the mobile users efficiently and in a real-time manner.

3) NETWORK TECHNOLOGIES

Multiple small cells are used in the MEC environment. Wireless network protocols such as WiFi and mobile cellular networks are the leading network technologies to provide connectivity between mobile devices and the edge server.

4) MOBILE DEVICES

Such devices can be used to process energy-saving and hardware-related tasks that cannot be offloaded to the edge network. Wearable devices also perform peer-to-peer calculations within the edge network using device-to-device communication.

5) SOFTWARE DEVELOPMENT KIT

This standard application programming interface (API) helps to adapt existing services, accelerate new flexible compute-intensive applications, and be easily integrated into the application development process.

C. MEC BENEFITS

In MEC systems, the computing resources are not centralized at the conventional cloud servers. They are distributed at the mobile network edges near the end-users. MEC infrastructure may include tens of large data centers and thousands of small ones collocated with cell towers and separated by less than 10 miles [25], as shown in Figure.2.

This scalable and flexible architecture allows MEC to offer the benefits below:

1) HIGHLY DISTRIBUTED AND HETEROGENEOUS RESOURCES

Edge DCs in MEC are allocated in different geographical locations, and they vary in scale and type in terms of computing resources such as processing resources, storage resources, and network resources.

2) SUPPORT REAL-TIME APPLICATIONS WITH LOW LATENCY

MEC is an excellent choice for services that require guaranteed QoS or low-latency communication, receive high traffic from end-user devices or need extensive data analysis.

3) LESS MOBILE ENERGY CONSUMPTION

In MEC, compute-intensive applications can be offloaded from the mobile devices to edge servers. Computation offloading will significantly reduce their energy consumption and prolong battery life.

4) MOBILITY SUPPORT

MEC can support mobile end users, including smartphones, IoT devices, sensors, and provide them with seamless access to the network by changing their attachment points to the network as they move.

5) IMPROVING SECURITY

The distributed deployment and the small-scale nature of the edge servers in MEC make it less vulnerable to security attacks. Moreover, MEC prevents uploading critical data to remote data centers. The IT administrator is responsible for maintaining the authorization and access control rules within the enterprise and categorizing various service requests without needing an external unit.

6) INTEROPERABILITY WITH THE TRADITIONAL CENTRALIZED CLOUD

In some cases, getting resources from the distant centralized clouds with much greater computing and storage capabilities is cheaper than MECs. So MECs should exploit resource allocation techniques to allocate end-users with computing resources with less latency, cost, energy consumption, and improved performance.

D. MEC POTENTIAL CHALLENGES

In MEC systems, despite the various opportunities that MEC can offer, several potential challenges are needed to be

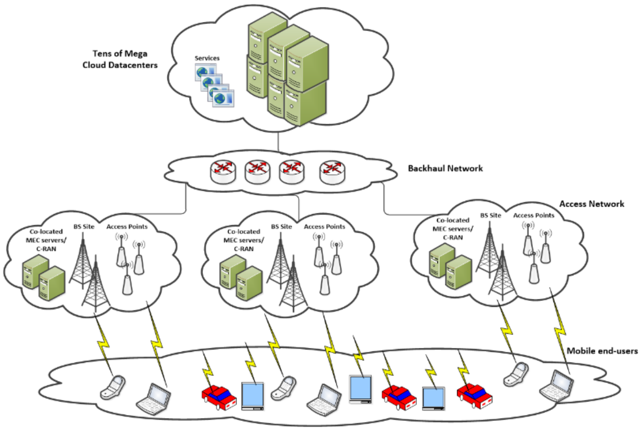

FIGURE 2. General MEC Architecture.

investigated to allow network players such as mobile end-users, infrastructure providers, and mobile network operators to get an advantage from the edge services:

1) DISTRIBUTED RESOURCE ALLOCATION MANAGEMENT

A multi-criteria resource management technique is required to handle the dynamic behaviors of the MEC system due to devices mobility and the regular change in the computing application requirements [26]. Implementing a multi-objective resource management technique requires to be combined with multi-criteria schedulers. This requirement could be challenging due to the variable application types, heterogeneous MEC servers, various user requirements, and the different QoS requirements of the communication channels. Furthermore, the wireless channel would become congested with the massive increase in mobile devices, and users would compete fiercely for the limited computing resources. This issue could be resolved using the centralized approach, but it has one drawback: high computing complexity and high reporting overhead. As a result, the centralized process is not convenient for distributed MEC systems [27]. Based on that, reliable and distributed MEC resource allocation schemes needed to be investigated.

2) SYSTEM INTEROPERABILITY AND APPLICATION PORTABILITY

Depending on the users' locations and the technical requirements, physical nodes can be deployed at different locations within the MEC infrastructure. As a result, a new crucial challenge is MEC's transparent integration into the underlying existing infrastructure and interfaces without affecting the standard specifications of the core network and end device. According to [28], the ability of the MEC system to communicate with other system elements in the 5G network to manage users' workloads and get appropriate control information is a critical component of MEC integration. Furthermore, the application migration points to a requirement known as application portability to eliminate the need for the software designers to create multiple instances for various MEC frameworks.

3) RELIABILITY AND MOBILITY

Managing mobility and ensuring reliability in a dense environment is extremely difficult. First, when multiple small servers are used, user mobility can lead to frequent handovers, service outages, and overall poor network performance [29]. Second, mobile users can change their positions during the computation offloading time (e.g., vehicles, mobile devices). In such a case, mobile users may be unable to access the computation outcome after processing their request as they left the service area of the home serving node. As a result, reliable computational offloading methods are required for the task computation's success taking into account the dynamic changes in the number of offloading users [30]. Third, considering the dynamicity of wireless connections and user mobility, providing efficient edge computing services in mobile environments is extremely difficult. For example, real-time applications like Augmented Reality (AR) necessitate real-time response and reliable connectivity between the edge nodes and end-users. Furthermore, these requirements would not be fully met due to dynamic channel quality and intermittent connectivity.

4) THE SYNCHRONICITY OF DECENTRALIZED MEC AND CENTRALIZED CLOUD

The conventional centralized cloud datacenters, with massive computational and communication resources, can process big-data applications in real-time and serve many users. However, the decentralized MEC infrastructure is highly desirable because it meets QoS user requirements and reduces latencies caused by traffic congestion and transmission delay. It is advantageous to implement MEC hierarchically, that is consists of three main layers: mobile user, edge-computing, and cloud-computing layers, like the HetNet architecture. The MEC provider also adds computational resources to the smaller eNBs, enabling HetNets to diversify radio transmissions and spread computing demands [31]. It is observed that decentralized MEC may not have enough computing resources to process all user requests, raising concerns about providing latency-critical services. As a result, it makes sense to distribute the latency-critical computing applications to decentralized MEC servers while moving compute-intensive and delay-tolerant applications to the conventional remote cloud [31].

5) SECURITY AND PRIVACY

MEC system has security and privacy difficulties. First, MEC can coexist with various network equipment, so using traditional privacy and security mechanisms used earlier in conventional cloud systems is inappropriate for MEC systems. Second, malicious leakers can perform overhead computation tasks. Thus, task offloading over wireless channels can be unsafe. The encryption on the user side and decryption on the destination server-side should secure the transfer of compute-intensive applications. However, this security procedure can increase the propagation and execution delays and reduce application performance [32].

Despite all these challenges, there is a potential need for a Mobile Edge Computing system in many critical real-life applications scenarios. The rest of this paper will focus on the entire process of MEC implementation, assuming there is no existing infrastructure deployed. This process includes reviewing and comparing current contributions to MEC infrastructure design and dimensions. Also, examining different works in MEC infrastructure virtualization, identifying virtual network services concept, VNF placement and autoscaling methods, and assessing the feasibility of the MEC framework. Moreover, the capabilities and performance approaches to optimize MEC infrastructure and resources are reviewed. Finally, an original implementation method is proposed to deploy an auto-scaled proactive MEC infrastructure at the mobile network edge.

IV. MEC DESIGN AND DIMENSIONING

MEC design and dimensioning process is defined as identifying the MEC edge node location and indicates the amount of computational resources required to be installed at each edge node based on the mobile traffic received at each site.

In recent years, there have been some related works on the placement of cloudlets [41]–[43]. For instance, Jia *et al.* [41] proposed cloudlet placement and mobile user allocation algorithm in the wireless area network. They presented two heuristic algorithms for the K-cloudlet placement problem: (1) a simple heavy-AP first algorithm that places cloudlets at the access points where the users' number is the largest. (2) a density-based clustering algorithm that eliminates the drawbacks of the heaviest-AP algorithm and decreases the probability of oversaturating a densely populated region with cloudlets. Xu *et al.* [42] examined the issue of cloudlet deployment in an extensive wireless metropolitan area network, addressing a new capacitated cloudlet placement problem to select the strategic locations in the WMAN network to place K cloudlets. The placement decision depends on the resources demands received by all user requests at a specific location. Xu *et et al.* solved this problem using a heuristic algorithm to reduce the average communication latency between mobile users and the cloudlets that serve the users. Xiang *et al.* [43] proposed a GPS-assisted adaptive cloudlet placement method for mobile applications. In this method, the cloudlet locations are identified based on the gathering areas of the mobile devices. Placing such mobile cloudlets improves the cloud service quality for dynamic context-aware mobile applications. Although the above methods are effective, they did not consider the workload of edge clouds in mobile edge networks or the delay in communication of remote users.

Some studies focused on MEC dimensioning topic by identifying the computational and communication resources demanded at each edge node to satisfy the system's workload. Some of these studies [11], [44] modeled the needed resources at each BS location as the number of MEC nodes that should be deployed, while other studies [45]–[47] modeled the required resources at each site as the workload of user requests. Wang *et al.* in [44] formulated a resource size problem to calculate MEC node locations and balance end-user communication latency and workload. MEC node locations are limited to base station locations. The resulting problem is NP-hard and is solved with an optimization solver. Takeda *et al.* in [11] addressed placing a fixed number of MEC nodes problem in mobile networks with a set of BSs associated with this edge node to offload computational tasks. They formulated four resource size sub-problems with various tradeoff targets between different performance metrics, including the maximum load of a single network link, the entire workload of a single MEC node, and the maximum network traffic. Li *et al.* [45] proposed a high-level approach to resource dimensioning where each BS collects workloads from its users, and these workloads are not tied to any specific type of computational resource. The authors in this work formulated the problem of resource dimensioning for placing MEC nodes to reduce energy consumption. The consumption of the system arises from two sources: (1) energy usage that occurs even when the server is idle, and (2) energy usage caused by the hosting user application. They, therefore, proposed an algorithm to maximize resource utilization and reduce the energy consumption at the MEC node. Authors in [46], [47] provided exemplary system usage models based on traffic coming from base stations or user requests. Zeng *et al.* in [46] assumed that the computational effort at each BS is proportional to the amount of its network traffic, which can be determined from historical data. They examined the problem of minimizing the number of MEC nodes deployed while imposing restrictions on communication latency and the number of BS that a MEC node can serve. Wang *et al.* in [47] modeled each BS site as the arrival rate of workloads and characterized each workload as the number of requested resources in terms of CPU cycles, memory size, and disk space. They propose a three-step algorithm which: (1) selects the locations of the MEC nodes to reduce the number of used MEC nodes, (2) calculates the number of resources to be placed on each MEC node by dividing the resources on the MEC nodes and adjusts the number of resources based on the actual workload, and (3) determines the connection capacity of the network. Note that the authors of this work did not consider the network latency requirement when dimensioning the MEC system.

Authors in [48], [49] considered network latency requirements when taking MEC dimensioning decision. For example, Ta *et al.*, in [48], solved the server deployment problem to increase the number of users that fit the delay requirement. They proposed a technique of placing the edge servers in high user density locations and developed two heuristic solutions to achieve that. Kasi *et al.* in [49] studied the MEC dimensioning problem to minimize the latency between MEC nodes and BSs, suggesting that each BS is assigned to a MEC node to obtain the required computational resources. They proposed three algorithms based on genetic and local search heuristics to solve the dimensioning problem.

Other works considered the users' mobility when solving the dimensioning problem. In [50], the authors assumed that the user mobility would result in a certain known amount of workload to be shifted from one MEC node to another. The authors calculated an upper limit on the maximum supported number of VMs that can be transferred from one MEC node to another, based on the available computing resources to meet the resource demand for service migration.

Considering the reliability parameter in MEC dimensioning problem, the authors in [22] investigated the problem of assigning a fixed number of MEC nodes to serve users in the local area. They advocated using the reset power metric as the number of MEC nodes that each user can access and formulate the resource size problem as a multipurpose integer programming (IP) problem. The resulting problem is an NP-hard problem and is solved by an approximation algorithm to maintain system reliability.

Those above dimensioning and design MEC methods do not follow an auto-scaled and proactive approach when determining the required number of MEC nodes and resources allocated for each mode to serve the user request. The MEC system's workload is assumed to be fixed in these works, which is not the case in the real environment. Static dimensioning approaches may lead to MEC resources not being utilized effectively over time and could also cause the blocking requests rate to increase as there are not enough resources to process the user request at a specific time. To efficiently deploy a MEC system at the network edge, the operator needs to find the proper server placement and the actual amount of communicational and computational resources required to process user requests dynamically varying over time. Thus, the operator should virtualize the MEC infrastructure using Network Function Virtualization (NFV) technology.

In an NFV environment, MEC network functions are implemented as Virtual Network Functions (VNFs), which can be deployed and executed on a virtualized platform (i.e., virtual machine or container). These VNFs can be easily placed at the hosted edge nodes and close to the end-user to process their requests. Placing VNFs at the right edge nodes and assigning them the proper amount of resources should be carefully managed by the operator considering the edge node capacity limitations, service latency requirements, and time-varying traffic. Some works related to VNF auto-scaling and placement in the MEC system are presented in section V.

VNF placement and orchestration could be challenging for the operator due to the mobile traffic variation based on user distributions and mobility models. This emerging behavior will result in uneven traffic distribution within the MEC system. Thus, the number of required VNF instances to be placed at the edge server will fluctuate frequently. To overcome these challenges, the operators should adopt a proactive auto-scaled approach to place VNFs in a distributed MEC-NFV infrastructure dynamically, allocate/release resources to a VNF automatically, and add/remove one or more VNF instances. The proactive VNF auto-scaling approach combines traffic prediction and threshold-based methods to produce scaling decisions ahead of time. Section V discusses different VNF auto-scaling and placement approaches used in dimensioning the MEC framework. Section VI provides a high-level description of the necessary steps to deploy the MEC-NFV infrastructure from scratch that proactively scaled the virtualized services at the distributed edge nodes in MEC according to the received users' traffic.

V. MEC FRAMEWORK

In the MEC system, the distributed servers in the infrastructure layer must provide the end-users services in a cloud-like manner, which means users should be unaware of such decentralization and distribution. As a result, the distribution of MEC servers over the physical infrastructures must be appropriately managed and orchestrated to offer service in a unified manner. Two factors should be considered to achieve this requirement when implementing the MEC framework: (1) Proper management for the server distribution in the MEC infrastructure, (2) Resource Orchestration over the distributed heterogeneous infrastructure, (3) Building an agile MEC framework that supports mobile traffic dynamicity using VNF auto-scaling and placement approaches, and (4) Optimizing the performance for the MEC framework. This section reviews the research proposals for managing MEC servers' distribution and resources among multi-sites.

A. MEC EDGE SERVERS DISTRIBUTION MANAGEMENT

MEC edge servers' distribution in the MEC infrastructure is not suitable to co-work with the existing cloud computing's united service model. Fortunately, Software Defined Networks (SDN), a newly emerging control technique, guarantees edge computing requirements, and it can offer centralized orchestration and virtualization capabilities over decentralized platforms [51]. While reviewing this current topic, one can notice that SDN could work with the MEC in three ways: (1) SDN can be used to control the edge to provide the unified service to its users; this approach is called SDN-Controlled edge, (2) Edge computing nodes can extend their infrastructure capabilities to implement SDN controllers at each node and reduce the response time for control messages and reduce delay; this approach is called Edge-Enabled SDN, and (3) Distributed computing edge elements and the SDN controllers can work together to manage and control users' requests; this model is known as SDN Collaborate with Edge.

1) SDN-CONTROLLED EDGE

Jararweh *et al.* [52] introduced a software-defined networking framework to provide MEC efficient storage and computing services. The centralized management nature of SDN conforms to the decentralized nature of edge computing; therefore, SDN can be improved to manage and control computing resources at the network edges [52]. From a system standpoint, this improvement is accomplished by applying global and local software-defined controllers responsible for ensuring that all edge features in the MEC system are working efficiently. Xu *et al.* [53] proposed a prototype of an edge computing node based on the SDN concept. The SDN controller suggested in this work allows edge servers to control functionalities and provides a reliable platform for performing critical data analytics required at the IoT source. Abdullahi *et al.* [54] developed and implemented an SDN-based edge computing control platform to coordinate clustering in Information-Centric Networks. Targeting the Internet of Vehicles (IoV), Hou [55] proposed using vehicles as infrastructures for communication and computation resources. In this work, SDN was augmented with the ability to manage various distributed vehicular resources and characteristics such as devices capabilities and mobility in the vehicular edge computing systems.

2) EDGE-ENABLED SDN

MEC broadens computational infrastructures and capabilities, providing more options for implementing SDN controllers as software instances on these infrastructures. For example, Liu *et al.* in [56] proposed an SDN-enabled network architecture aided by MEC that integrates various types of access technologies. SDN control components can be spread to the physical MEC infrastructure using the proposed architecture to minimize control message response time. More specifically, the control plane can get device location, speed, direction, and accessibility to the network in real-time in such an architecture.

3) SDN COLLABORATE WITH EDGE

MEC is built on distributed IT elements, and SDN was created to manage and control computing resources over the edge network. Depending on this connection, they can collaborate to accomplish a common goal. For example, Aggarwal *et al.* in [57] proposed a collaboration model to ensure IoT devices' security with SDN and edge computing. This model considers the intelligent SDN networking paradigm in network device reconfiguration, traffic rerouting, and applying authentication and access rules that can open the way for improved security.

B. MEC RESOURCE ORCHESTRATION

In addition to centralized control of distributed MEC infrastructures, computational and communication resources in such infrastructures must be managed to provide cloud-like network services to the end-users. This subsection presents the existing work of resource orchestration from two viewpoints: load balancing and multiple edge servers' collaboration.

1) LOAD BALANCING

Load-balancing can be viewed as distributing the workload among edge data centers to make the operations more efficient by avoiding congestion, low load, and overload. Users' traffic can be dispatched to various edge servers to accomplish load balancing in MEC, which offers many edge servers as candidate service points. Because users' workloads change over time, the load distribution among different MEC servers may need to be adjusted accordingly. Users' traffic can be dispatched based on the balancing strategies and objectives. For instance, Song *et al.* in [58] proposed a dynamic graph partitioning approach to assign the workload of each edge DC. The main goal of this approach is to have a dynamic load balancing method that can efficiently minimize intra-DC migration and thus minimize the node migration consumption caused by the continuous MEC system changes. An example of the dynamic graph-repartitioning algorithm was proposed in [63]. This algorithm takes an input prior load-balancing results and reduces the gap between the load-balancing result and the initially proposed status. The authors focused on deploying an effective dynamic load-balancing algorithm with an authentication method for edge servers [63]. Each edge node is represented using the current workload and the maximum allowed resource capacity required to process this workload in this work. Based on the proposed system and using Breadth-First Search (BFS) algorithm, tasks are assigned to an under-utilized edge node. The authentication method allows the load-balancing algorithm to find an authorized edge node. Targeting the mobile networks, Fernando *et al.* in [59] proposed a work-sharing model known as Honeybee. It is used to balance the workload of independent tasks among various edge nodes. This model can handle the variation in user load as users' locations change and accommodate mobile nodes randomly leaving and joining the system. In [60], Oueis *et al.* proposed an algorithm to deploy small cell clusters and customize the resource management to optimize the complexity of load balancing algorithms for edge computing. This complexity is caused by the dynamic computations and the frequent changes in the load condition among the edge nodes.

2) MULTIPLE EDGE SERVERS' COLLABORATION

MEC servers are usually located near the mobile end-users to ensure low latency and high QoE services. Therefore, the MEC servers' capacity is generally limited to manage infrastructure costs. Consequently, one MEC server cannot work independently to serve all users' requests, and it requires cloud DC support when necessary. Multi edge server collaboration in MEC could be classified into two categories: Horizontal or Vertical. Tran *et al.* in [61] proposed the horizontal collaboration approach, which coordinates IT resources within the same layer. This approach is helpful to achieve load balancing, system reliability, and failure recovery. This failure recovery scheme, which focuses on user device collaboration, is intended for situations in which no nearby MEC servers are accessible within the transfer range. In this case, some close mobile devices will be selected as ad hoc relay nodes and connect the affected user devices to unreachable MEC servers. Cardoso *et al.* in [62] proposed the vertical collaboration approach that coordinates mobile devices, physical edge servers, and remote cloud centers for performance optimization. They extended the current edge computing software stack to allow efficient collaboration and orchestration. In this work, external drivers are designed by a company to support new network equipment in an edge-computing system. Test results show few drawbacks of the proposed solution in terms of latency and throughput and that deployment times are shorter than similar maintenance operations on traditional computer networks.

C. MEC AUTO-SCALING APPROACHES

The three-tier edge architecture previously shown in Figure 2 has demonstrated MEC benefits such as the reliability and availability of the local and global services. However, the MEC framework is being challenged by three major trends: the rapid growth of the data generated by end-users, latency-critical requirements by modern computing applications such as augmented reality applications [64], and the dynamicity of traffic patterns in 5G mobile networks. Considering these challenges, many works were conducted to find an intelligent edge between mobile devices and centralized cloud architecture. That could be achieved by virtualizing the underlying MEC infrastructure using the NFV technique. In the MEC-NFV architecture framework, the MEC network functions are isolated from their underlying hardware, and they are deployed as virtual network functions and on VMs or containers within the MEC edge server. To support the varying workload dynamics considering the edge node capacity limitations and service latency-critical requirements, mobile network operators should deploy and manage VNF auto-scaling mechanism at its edge nodes. Existing research on VNF auto-scaling can be split into two categories: Proactive approaches and Reactive approaches. The following subsections discuss each mode and list the current research related to it.

1) REACTIVE AUTO-SCALING APPROACH

In the reactive auto-scaling approach, the threshold levels are statically pre-defined and are easily implemented. The thresholds are hard to choose to handle the dynamic traffic in 5G networks. In [65]–[68], the authors proposed VNF auto-scaling mechanisms based on static thresholds (lower and upper scale thresholds). The auto-scaling process is triggered if the traffic load falls below or exceeds these thresholds. For example, the proposed VNF auto-scaling technique in [65] aims to utilize cloud resources and improve QoE cloud services efficiently. Carella *et al.* in [66] presented an Autoscaling Engine (AE) mechanism that dynamically adapts the network services provided by Telecom operators and increases the reliability, stability, and resource utilization of these services. Authors in [67] proposed threshold-based auto-scaling for VMs to dynamically scale the virtual instances based on the application computing resources utilization.

Similarly, in [68], Hung *et al.* introduced an auto-scaling algorithm to enable the automated provisioning and balancing of VM resources depending on the active session of the end-user application. The techniques mentioned above could lead to an oscillation in the scaling decision and, thus, unstable behavior that may affect overall system performance. Alternatively, [69] and [70] introduced queueing theory and reinforcement learning mechanism into threshold-based auto-scaling techniques to improve the system performance. Still, it stays reactive approaches with the same weaknesses.

2) PROACTIVE AUTO-SCALING APPROACH

In proactive scaling approaches, forecasting techniques allow systems to automatically learn and predict future demands on which auto-scaling decisions are made.

Machine learning (ML) is considered one of the leading prediction techniques used to forecast the future workload for auto-scaling. ML introduces intelligent network operations into the mobile edge architectures as the network system automatically achieves its goals without human interference. The system reacts and responds to any change in its environment and learns from experience based on the historical statistical information gathered during its operation. This ML intelligence will minimize costs and errors, increase system efficiency, and reduce resource management in scalable networks.

ML has the potential to expand the boundaries of what MEC can achieve, allowing for the construction of more complex systems. Such systems, in turn, provide more computing resources and better potential of delivering high QoS. MEC with ML, for example, makes the use of mobile and satellite servers possible, ensures more collaboration between servers in different networks, and predicts the network status. [71] also highlights which various ML solutions can solve MEC computation and communication challenges.

Machine Learning plays a dual role in the context of edge computing which was presented in [72]:

- ✓ ML introduces new capabilities to optimize MEC infrastructure: Distributed control and system orchestration, predictive network infrastructure maintenance, initiation of MEC resources, and proactive application offloading decision depending on the workload on each MEC node or end-user mobility across the MEC system.
- ✓ Training ML: distributed edge platforms promptly provide massive amounts of local data to ML models. ML gets the input from the underlying edge platform to enhance its operational efficiency. This topic opens up novel research directions to explore ML optimization approaches based on the workload information received from the edge system, including predicting the changes in the workload at the edge nodes to orchestrate the computational resources and task offloading in runtime. The most effective ML-based optimization approaches of MEC infrastructure are listed in the following.

In the context of proactive VNF auto-scaling in a distributed NFV-based edge infrastructure, authors in [73], [74] proposed a dynamic mechanism like reinforcement learning to enhance NFV scaling policies based on dynamic thresholds. As it outperforms static approaches, it is still a reactive solution with the same flaws. For instance, Arteaga *et al.* [73] proposed an adaptive scaling mechanism for NFV based on Q-Learning and Gaussian Processes. An agent uses it to manage performance variations better and implement an effective scaling policy strategy. This mechanism was validated through simulations in a virtualized Evolved Packet Core (EPC) case study, proving that it is more accurate than approximations.

On the other hand, [75] proposed a time series model to predict resource usage based on a historical dataset. Other authors, including Mijumbi *et al.* [76] and Mestres *et al.* [77], used ML models to handle the fluctuation in managing the amount of the virtualized resources assigned to specific VNF instances by anticipating resource requirements and thus improving the resource allocation algorithm efficiency. For example, a neural network-based model is used to predict the future resource requirements for each VNF instance based on VNF forwarding graph topology information (VNFC) has been proposed in [76]. Each VNFC's topology information is derived by combining its previous resource utilization and the modeled effect on its neighborhood. The proposed approach was evaluated for real VoIP traffic traces, and the results showed a reduction in the dropped calls rate and an improvement in the call setup latency.

D. SERVICE PLACEMENT IN MEC

Service placement in the MEC system is about to place the virtual network functions (VNF) on the available MEC nodes. Each VNF is a service instance that processes the end user's request. The service placement could be subject to available computational and communication resources based on the service type and the use case. VNF placement algorithms could provide three service types: (1) latency-sensitive services, (2) services for mobile users, and (3) service chain. Finally, the VNF placement problem is optimized to reduce the operation cost, resource utilization, service latency, and/or energy consumption. A review of these optimization approaches is discussed in this section. This subsection presents the used placement algorithms policies and available services for VNFs.

1) VNF PLACEMENT POLICIES

Some VNF placement proposals mainly focused on meeting the computational resource requirements when placing the VNF [78]–[81], [14]. In contrast, other proposals focused on allocating communication resources in either the wireless network [82], [85] or the wired network (e.g., a mobile backhaul network) [86] to provide services with high data rates in the MEC system while performing the placement of the VNF.

a: VNF PLACEMENT BASED-ON COMPUTATIONAL RESOURCES

In [80], the authors formulated a service placement problem to maximize the rewards for the service to the users. The authors showed that the proposed service placement problem is NP-hard and then presented an approximation algorithm using the set cover problem. In [79], the service placement problem was investigated to reduce the total cost of computing and communication resources. The authors proposed a heuristic based on a genetic algorithm. Since MECs are closely linked to cellular networks, it is interesting to investigate whether the combined consideration of service placement and RAN design can facilitate the integration of cellular networks and MEC systems. This problem was examined in [81] to compute a Pareto-optimal solution for network cost and service delay. The problem in question is NP-hard, and the authors proposed an algorithm based on Bender's decomposition to calculate approximate solutions. The algorithm breaks down the general problem into a master problem for computing service placement and a sub-problem for computing RAN configuration. The solutions for the master problem and the sub-problem have a lower and upper bound of the resulting solution, and the algorithm stops when the upper and lower bounds match. The results showed that the general approach could significantly reduce the overall cost compared to a non-optimized cloud RAN.

The authors of [14] suggested a decentralized service placement approach in which location decisions are made jointly by end-users and MEC service providers. MEC service placement is done in two stages. In the first stage, each user is assigned an algorithm to determine whether to perform tasks on a local server or MEC nodes, while in the second stage, the MEC nodes required to process the user service are computed. Then, based on the calculated workload and costs, the login algorithm determines the order of services be entered by the MEC service provider. Joslio *et al.* in [82] proposed a decentralized workload of the algorithm with a bounded approximate relationship and assumed that the end-users share communications and computing resources in the MEC system. In this approach, each user decides which MEC node will load their work to reduce the weighted amount of their power consumption and response time. In [83], the authors considered the same MEC system with users generating functions periodically. They proposed a decentralized algorithm that would gradually allow new users to make task loading and assignment decisions.

b: VNF PLACEMENT BASED-ON COMMUNICATION RESOURCES

In this context, we will discuss allocating the communication resources in both the wireless and wired networks. In wireless networks, allocating the communication resources allows end-users to get achievable data rates according to their needs. For example, the authors in [82] considered the possibility of hosting a group of services in the MEC to meet the mobile users' requirements, considering the computational and communication capabilities of MEC nodes to reduce traffic congestion. In particular, the throughput of the MEC node is viewed as the number of available resource blocks (RBs), which meet the individual data rate requirements. Two sources are required to host a particular service: the first part is the power consumption of the main activities of the service, while the second part is directly proportional to the number of mobile users served. This paper proposed a greedy algorithm of a single MEC node where each node selects the services to be hosted based on their resource efficiency and utilization. Moreover, it presented a decentralized plan based on mapping and mutual preference between MEC nodes and services. The results showed that a collaborative approach to service and radio resource deployment reduces traffic congestion and significantly improves computing resource efficiency. Authors in [85] investigated the allocation of wireless communication resources for the offloading requests in MEC. They modeled the interaction between BS and end-users as a Stackelberg game and proposed decentralized approximation algorithms concerning different operator resource allocation policies to minimize the completion time of task offloading.

When discussing allocating resources in wired networks like mobile backhaul, joint consideration of service placement and communication resource allocation was proposed [86]. This approach facilitates meeting end-user demand and is used to place VNF in MEC. The authors formulated a multi-criteria problem to reduce the implementation cost and balance the workload on network links, subject to the latency requirement of individual services and available computing resources. The authors expressed the resulting problem as a Mixed Integer Linear Program (MILP) and proposed an effective heuristic algorithm.

2) VNF SERVICES

VNF placement in the MEC system could be used to provide services that are impacted by the latency or user mobility or providing a service chain that should be processed by multiple VNF as follows:

a: LATENCY-CRITICAL SERVICES

In MEC, service delay is impacted by the end-users locations and the MEC nodes hosting each service. Many studies [15], [16], [87], [88] have been conducted to solve the VNF placement problem with a target to minimize the latency for the critical services. The approach in [16], [88] solved the VNF placement problem to support latency-critical services by prioritizing the delay performance overall performance measurements, which minimizes delays as the primary goal of the VNF placement. In this approach, the service latency can be expressed as the summation of MEC nodes' communication and processing latency. For example, the authors in [16] intended to place multiple VNF instances to serve a set of users, considering the MEC node capacity. Authors in [88] modeled a MEC system with access to central clouds. The authors suggest that it is better to allocate users' requests that need intensive resources on the cloud DC instead of the MEC system. The E2E delay should include the significant cloud delays in such a case. The authors formulated the NP-hard placement problem to reduce the average latency of service. The authors proposed a brute-force algorithm that lists all possible placements to calculate the optimal solution. In these two approaches, the latency performance of the user services may not be guaranteed as their main objective is to minimize the overall latency of the edge computing system only without any threshold. This issue could be solved by taking the latency threshold of the user service as a constraint to solve the VNF placement problem. For instance, authors in [15] followed this approach to increase the number of hosted services in the edge computing system. The authors first proposed an algorithm to calculate the best solution without limiting the capacity of MEC nodes. In the case of qualified MEC nodes, the authors proved that the problem is NP-hard and followed the approximation approach to solve it. In [87], the authors observed the same approach while considering deploying a service to balance the workload between MEC nodes, subject to MEC node capacity limitations. The problem is proven to be NP-hard. Using the tabu search, the authors have suggested a solution, starting with the earliest possible solution and gradually improving the proposed solution by changing service events. The algorithm ends when the maximum number of rounds is reached.

b: MOBILITY SERVICES

Due to the mobility of MEC users, adjusting VNF placements based on users' real-time locations can enhance the user experience [89]. A common approach has been used by different studies [90]–[92] to solve the VNF placement problem for mobile users. This approach divides the time into successive time slots and assumes that the connection between mobile users and BS within each slot is stable, and the user moves within BS coverage. Therefore, service placements for each location slot can be calculated, and services can be switched between different time slots to follow the user. In [90], the authors formulated the service placement problem with user mobility to minimize the average response time of the system. The resulting problem is an integer nonlinear programming problem, and the authors proved that the problem is NP-hard. The first step uses a genetic algorithm to determine whether service migration is necessary based on current system delays and latency caused by service transfers. If the service is decided to be migrated, then a sub-problem, NP-hard, is used to calculate the new placement and is solved by optimization solvers. However, this work is not suitable for latency-critical services, as it is not considering the user request's latency requirements. In [91], the authors considered hosting virtual reality (VR) games for dynamic user groups to reduce communication and computing costs and the latency of communication between users within each group. They considered the general pattern of user mobility in a time slot and proposed predictive control (MPC) method to estimate the number and locations of users for each time slot and then place the services according to the prediction, which is proven to be NP-hard. This placement problem could be solved using an efficient algorithm that defines an approximation limit for the service placement for each time slot.

Another method of capturing user mobility is to model the cost-of-service migration to track user movement. Migration costs are proportional to the amount of network traffic and computational resources. This method was used in the time slot system [92], where the authors assumed the general idea that migration costs are proportional to the user's workload and inversely proportional to the distance between users and MEC nodes. The authors formulated a joint problem to enhance QoE while reducing service migration costs. Other approaches considered a system with infinite time intervals, such as in [93], where an online approach was suggested to solve the service placement problem to meet the demand for end-user by reducing the weight of computational latency, communication delays, and transfer delays. The authors first developed a dynamic service placement problem as a contextual multi-armed bandit problem then suggested using Thompson sampling to estimate users' expected performance for various location decisions. The authors only considered communication delays as system parameters in this model and did not include bandwidth allocation.

c: SERVICE CHAIN

A service chain in MEC means that the end-user request must be processed by multiple services that depend on each other. Inter-service dependency adds a new dimension to managing service chain placement. For instance, the end-to-end latency of a user request in the service chain should include communication latency between the dependent services and traffic flow. Some constraints can be relaxed to make it easier to develop solutions to address the complexities of service chain placement. For example, authors in [94] investigated service chain placement problems regardless of the MEC node capability constraint. They proposed a heuristic based on a local search and Hungarian algorithm to reduce the total cost of computing and communication resources.

Additionally, authors in [95] worked on the service chain placement in MEC to balance the workload of nodes. The authors modeled each service chain as a graph, where each vertex and edge in this graph corresponds to the service or communication path. The resulting deployment problem is NP-hard, and the work suggested two solutions. The first solution focused on the best placement of a single service chain, and then the second solution took an online approach to place multiple service chains, each of which is a tree in the diagram.

Service chain placement can also be performed under the capability of MEC nodes, including available CPU cycles, memory size, and hard drive space. Works in [96], [97] modeled the capacity of MEC nodes in the service placement problem. For example, authors in [96] proposed service chain placement to increase resource utilization efficiency. They suggested a polynomial solution based on graphs and the Hungarian algorithm. Furthermore, the authors of [97] studied the problem of placing a service chain to reduce deployment costs. They proposed a heuristic solution based on the genetic algorithm and suggested two strategies for applying the proposed solution. One is to calculate the assignment of the service chain sequentially, and the other is to calculate the allocation of the service chain together. Simulation results showed that the latter strategy reduces the total cost and requires more execution time.

Other works solved the service chain placement for user requests with high traffic demand, where network traffic transfer paths should be calculated under network capacity, traffic load, and dependence between services limitations. This problem was studied in [98], where the authors considered the combined problem of service chain placement and flow distribution to optimize proper flow and energy consumption jointly. The resulting problem was expressed as an NP-hard model and solved using an approximate algorithm based on linear relaxation and estimation techniques. The proposed solution first calculates the placement of services and then calculates the optimal flow distribution. Statistical results showed that a combined computing and communication resources allocation is necessary to optimize efficiency for systems and services.

E. MEC OPTIMIZATION APPROACHES

ETSI primarily classifies MEC use cases into three categories: consumer-oriented services, operator-oriented services, and network performance-oriented services [99]. The MEC systems should support all these categories to enable a wide range of new services and computing applications at the edge nodes. In general, the categorization of the MEC use case depends on who could benefit from the application advantages. For example, the ''consumer-oriented services'' use case seeks to provide direct benefits to the end-users by enabling the execution of computation-intensive and latency-critical applications at the edge nodes. It could be achieved using the computation offloading method, where mobile devices can use a massive amount of computational resources on the edge nodes. Second, the ''operator-oriented services'' use case directly benefits the mobile operators and third parties to use the computational and storage resources at edge nodes to allocate their applications and services. Operators and third-party vendors' applications and services can include extensive data analysis, active device location tracking, security, safety, and data analytics. Finally, the ''network performance-oriented services'' use case aims to optimize network operations, improving network performance and QoE local content caching at the edge server like video delivery optimization for TCP.

MEC framework is designed and deployed in a way to be able to offer high QoS to its user. To fulfill the targeted performance requirements of the latency-critical applications offered by MEC, it should exploit its resources and infrastructure efficiently. Thus, several QoS parameters should be considered in the optimization approaches. Identifying which QoS parameters should be optimized depends on the component that will get an advantage from the MEC system, such as the service provider, the infrastructure provider, and the end-user. So, let us examine the current optimization approaches in the MEC system based on these three perspectives and which QoS parameter is being optimized.

QoS parameters examined in this section are Latency, Computing Resources (including processing, memory, and Bandwidth), Energy, and Combined Resources. Most of the works in this section used the VNF placement algorithm to formulate the optimization problem then solved it to meet the required QoS. Table.1 shows a summary of the MEC framework optimization approaches listed here.

1) OPTIMIZATION APPROACHES FROM INFRASTRUCTURE PROVIDER PERSPECTIVE

From the infrastructure perspective, the proposed optimization approaches are formulated and solved to benefit the infrastructure provider to either minimize the energy consumption of MEC servers, reduce communication latency, or utilize the computing resources efficiently on the edge nodes. For example, the energy consumption in MEC is primarily caused by MEC servers, network equipment, and user devices. To improve the energy consumption at the edge nodes, the authors of [100] proposed a power consumption-clustering scheme that minimizes MEC environment power consumption while keeping the average traffic processing time below the threshold. The optimization problem in this work has been solved to find the optimal number of clusters to reduce the power consumption of the MEC system. The average system power consumed is approximated as the average utilized CPU for all the edge nodes of the network. The average power consumed at each edge node depends on the number of requests received within the workload. One of the limitations of the proposed method is that each virtual machine deployed on an edge node can process the requests assigned to one virtualized network function at a time, making this solution unsuitable for large-scale MEC environments.

To implement an effective 5G service comprised of virtualized network functions, an energy-aware VNF placement strategy is needed. The authors of [101] designed cloud-enabled small cells in a 5G environment. The main objective of the proposed energy-aware VNF placement problem is to minimize the power consumed in the MEC system limited by network service latency requirements and infrastructure terms. The overall system power consumption depends on the power required by all VNFs to process their assigned workload. The power consumption of each VNF is identified by these three factors: 1) processing resources utilized of the hosting virtual machine; 2) small cell power consumption; and 3) power consumption of physical network equipment that switches traffic among different VNFs. Furthermore, the authors of [101] used a solid constraint to overestimate the allocated resources to predict traffic peaks. One of the limitations of this proposed approach is that each virtual machine can host only one VNF instance.

Authors in [102] investigated the joint offloading and autoscaling problem in energy harvesting MEC systems. They discovered that foresight and adaptability are critical factors to ensure the reliability of renewable-powered MEC operations. In this work, a reinforcement learning “RL” algorithm was developed to learn the optimal offloading and autoscaling policy in the presence of a priori unknown set of parameters. Compared to standard RL algorithms such as Q-Learning, the proposed scheme uses online and offline RL algorithms to improve learning and process runtime efficiency. The simulations revealed that the suggested model could effectively enhance MEC performance using sporadic and uncertain renewable energy. In the above work, each base station determines its offloading and autoscaling actions considering energy harvesting based on the workload received from end-users.

TABLE 1. MEC framework optimization approaches.

Ref.	Target Component	Optimization Objective
[99]	Infrastructure Provider	Minimize MEC environment power consumption
[100]	Infrastructure Provider	Minimize overall system energy consumption
[101]	Infrastructure Provider	Use Q-learning mechanism to reduce MEC system energy consumption
[102]	Infrastructure Provider	Reduce communication delay in MEC clusters
[103]	Infrastructure Provider	Minimize energy consumption and task execution time on MEC nodes
[104]	Infrastructure Provider	Enhance computing, storage, and bandwidth resources at the edge nodes
[105]	Infrastructure Provider	Minimize hardware deployment costs while maintaining the required communication delay
[106]	Service Provider	Reduce services’ latency and increase their availability
[107]	Service Provider	Minimize end-to-end communication delay and reduce service deployment cost
[108]	Service Provider	Reduce end-to-end service time in a MEC network
[109]	Service Provider	Minimize service operating cost and provide high availability services
[110]	Service Provider	Minimize service energy consumption
[111]	Service Provider	Reduce the cost of service migration and provide high availability services
[112]	Service-Provider	Use ML to utilize computing resources efficiently
[113]	End-user	Minimize latency and support real-time and critical-latency applications
[114]	End-user	Achieve service delay required by users in a MEC-based 5G network while minimizing inter-MEC handover costs
[115]	End-user	Provide high availability services to the end-users
[116]	End-user	Provide low-cost services to the end-users in a multi-tenancy architecture
[117]	End-user	Minimize energy consumption on the end device and reduce the task computing execution times
[118]	End-user	Improve energy efficiency and QoS for edge end-users
[119]	End-user	Support heavy computation tasks received from end-users while minimizing the energy
[120]	End-user	Provide tasks with low latency to the end-users

Authors in [103] proposed an optimization problem of VNF placement in a MEC environment to minimize the communication and network delay and find the optimal placements of VNFs. A dynamic resource allocation mechanism was used to adapt the current VNF placements to address time-varying workload when hosting edge nodes reached their capacity limits. This mechanism could predict the system’s future workload and identify which edge nodes will over-utilize their capacities. One of the scenarios intended here is that the current edge node cannot handle and process the anticipated workload as it violates the latency requirement. Thus, a new edge node should host VNF to process the workload within the latency threshold. This could be achieved by using the online adaptive greedy heuristic algorithm. This proposed algorithm determines the location of the new node while balancing the load to the new service-hosting nodes.

Authors in [104] proposed a task offloading approach using Q-learning to minimize execution time considering energy consumption constraints and the inherent dependency of the tasks. However, the work focused on demonstrating the flexibility and efficiency of the RL-based approach for MEC without considering more complicated MEC use cases involving multiple mobile devices.

Authors in [105] proposed a VNF placement scheme to avoid wasting edge servers' computational and communication resources, including processing, storage, memory, and network resources. Enhancing the resources utilization in the edge environment is the primary goal of the proposed NFV Chain Placement problem. This objective could be achieved by minimizing resource portioning on edge servers. The main point is to keep the available resources on each edge node as small as possible after the VNFs are deployed. A heuristic algorithm is used to identify the optimal new VNF places while reducing the available capacity of the hosting edge node after deploying VNF.

Authors in [106] proposed a VNF placement approach that facilitates the implementation and placement of service-chained virtualized functions in a cloud environment that can be extended to MEC infrastructure to provide services for mission-critical applications. The main objective of the VNF placement problem was to minimize hardware installation expenses while maintaining the required communication delay. The communication delay depends on the latency introduced by the traffic propagation, traffic transmission delay, traffic processing delay, and server queueing delay. According to the Tabu Search meta-heuristic, a suboptimal algorithm was also developed to get VFN placement solutions quickly. One of the drawbacks of the proposed approach is that it was not tested against compute-intensive applications' real-time characteristics and requirements.

2) OPTIMIZATION APPROACHES FROM INFRASTRUCTURE SERVICE PROVIDER PERSPECTIVE

Because MEC servers are geographically distributed, their location determines network transmission latency caused by the distance between MEC servers when fetching related services. Furthermore, given a set of MEC servers, the workload assigned to each MEC server determines the processing latency in the application layer. As a result, service latency is mainly identified by service placement and workload distribution. In this context, a VNF placement algorithm was proposed by Yala *et al.* [107]. From the service provider perspective, the latency of the service provided by the VNF depends on the access latency of the physical edge node that hosts the virtual machine. In comparison, the service availability is impacted by the reliability of the physical edge nodes and the hosting VMs. In [107], the VNF placement problem is formulated as a multi-criteria optimization problem with processing resource utilization and energy consumption constraints, limiting the cost of deployed services. The VNF placement algorithm solves the problem by balancing latency and the high availability of services. Lievadeas *et al.* in [108] proposed a novel placement and deployment approach for the service-chained VNFs in MEC infrastructure. The approach's primary goal was to reduce E2E delay supporting mission-critical and delay-sensitive traffic while satisfying the service providers' target of low deployment cost.

The authors of [109] proposed a clustered NFV service chaining technique to optimize total service time in a computing edge network. A deterministic algorithm was used to group virtualized functions based on the users' demands, determined by the probability of workload requests.

Efficient VNF placement minimizes the operation and installation costs while increasing MEC application availability. As a result, the authors of [110] investigated the VM placement problem as a randomized programming model reducing MEC service costs. A heuristic algorithm is used in the proposed model to create a trade-off between service availability and low bandwidth cost and make it possible to use this model in the real world. This trade-off could be achieved by hosting the optimum number of virtual machines to process specific workloads on the same physical node. Thus, reducing the cost of network bandwidth. The authors used these two assumptions when working on the algorithm: (i) the failure of a virtual machine is not impacted by the failures on other machines, and (ii) the failure of a physical edge node is unrelated to another edge node in the MEC system.

Liang *et al.* [111] proposed an optimization problem to optimize energy consumption in a MEC and caching network. To minimize energy consumption in the MEC system, link bandwidth availability and content source selection parameters should be determined. Based on the proposed model, the energy consumed at each edge node depends on the operation and transmission energy required to process the offloaded work. The operation energy is calculated as the sum of the power consumption needed by the computation resources, electric circuits, and control signals. In contrast, transmission energy is determined by wireless and backhaul traffic. Several limitations were determined to ensure that the allocated resource of links does not exceed the backhaul bandwidth and radio resource. However, this approach presents some drawbacks because the following parameters were not considered: (1) the user mobility, (2) the handover, and (3) the energy consumed by the edge server.

In [112], the authors proposed an optimization problem to fit MEC applications' latency and reliability requirements from a service provider perspective while reducing the service migration, processing capacity, and bandwidth expenses. If the main ones fail, the application reliability is guaranteed by assigning the available computing resources and bandwidth at the backup servers and physical links. The handover rate among cellular cells is also considered when solving the optimization problem to meet the critical-latency applications requirements. A set of heuristic algorithms for allocating resources and efficiently routing users' workload to data centers was presented to address this issue. Yet, this approach does not consider the time-varying traffic patterns

Authors in [113] discussed the optimal utilization for computational resources in MEC systems using ML from the service provider perspective. ML can be considered as an effective solution in such a complicated scenario where the number of sophisticated devices grows rapidly, and the number of configuration parameters becomes larger accordingly.

3) OPTIMIZATION APPROACHES FROM AN END-USER PERSPECTIVE

From an end-user perspective, optimization approaches focus on supporting the execution of critical and intensive computing applications and services at the edge nodes. For example, Cziva *et al.* in [114] investigated the VNF allocation problem on various edge nodes to reduce the total approximated end-to-end delay between mobile devices and their assigned VNFs. The proposed scheme automatically allocates VNFs on the physical nodes depending on the delay variations, heterogenous users' requirements, and device mobility. The optimization goal of this work was to enable running latency-critical applications at the edge nodes. One drawback of this paper is that the authors assumed that their proposed approach's total migration cost is time-independent, which is not the case in real environments.

Chen *et al.* in [115] proposed a model for the inter-MEC handover problem to accomplish service delay needed by mobile devices in a MEC-based 5G network. Their model determines the initial places of VNF, source and destination edge nodes, when and where VNFs should be transferred from one edge node to another, and the amount of computational resources should be kept aside on each edge node. The issue is formulated as an optimization problem to minimize the overall handover cost. A heuristic algorithm was used to solve the problem in the MEC server while maintaining the required sequencing order and the service latency limitation.

The high availability of the applications provided to end-users by the MEC system is guaranteed if the required virtualized functions responsible for processing the application requests are hosted and implemented on a wide range of heterogeneous edge nodes. Because of the computational resources constraints in the MEC system, [116] ensures the computing applications' availability using redundant VNFs. Implementing VNF redundancy was formulated as an optimization problem that reduces resource utilization costs. In this work, the failures of hosted virtualized network functions are unrelated to the other failures in the MEC system since the network functions are configured separately. This proposed approach cannot optimize scalable MEC infrastructure's performance that could support multiple real-time applications and services chains related to each other and require more complicated resource management and orchestration system.

The authors of [117] proposed the VNF placement problem considering the service function chains (SFCs) requests received from multiple users located at various positions in a hierarchical and geo-distributed architecture. The main goal of the optimization problem is to find the optimal VNF places while reducing the overall utilization cost of the computational resources by finding a tradeoff between the consumed bandwidth and computational resources. This could be accomplished by placing VNF requests with the same type on the same virtualized function instance, and thus it minimizes the number of deployed virtualized function instances on the edge node.

The authors in [118] investigated 5G heterogeneous networks to improve energy efficiency when offloading computing applications. Each mobile device decides whether to offload its task to the edge node or process it locally based on the energy consumption. In this case, the total energy consumption is determined by the transmission energy and processing energy. The proposed optimization problem reduces the overall service energy cost while considering latency-critical applications requirements.

In [119], the authors presented an energy-efficient Device to Device'' D2D edge computing offloading architecture. Both D2D and cooperative relay-aided transmission techniques are used to offload the computational applications and the wireless traffic received from the end-users in a multi-cell scenario. Traffic offloading and balancing technologies improved overall energy efficiency and service quality to edge users while reducing inter-and intra-cell interference and computational congestion.

The authors of [120] considered the case of a MEC system with multiple smart mobile devices that acquire compute-intensive applications from edge nodes. They proposed a new model for jointly optimizing offloading selection and radio and computational resource allocation. The optimization problem was formulated as a mixed-integer nonlinear programming (MINLP) problem to improve energy consumption efficiency considering the latency limitation. The authors proposed a reformulation-linearization-technique-based Branch-and-Bound (RLTBB) method to solve the optimization problem that ensures at minimum a suboptimal solution. Note that their optimization approach considers only the energy consumption without evaluating the latency or computing resources required.

Wang *et al.* in [121] proposed a multi-stack RL algorithm for MEC to allocate computational resources efficiently to process users' requests. In this algorithm, each BS can store historical user information and resource allocation schemes to prevent learning the same resource allocation scheme while enhancing convergence speed and learning performance. Compared to the standard Q-learning algorithm, the delay among all users can be reduced by up to 18%.

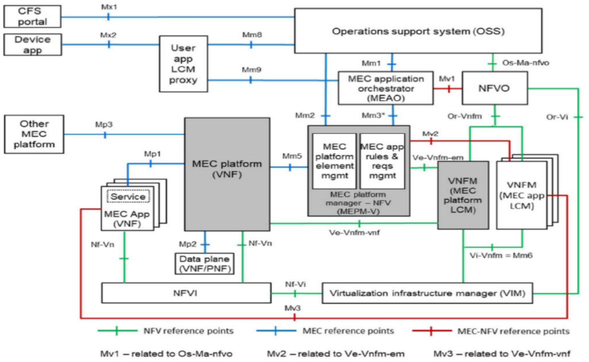

FIGURE 3. ETSI MEC-NFV Reference Architecture.

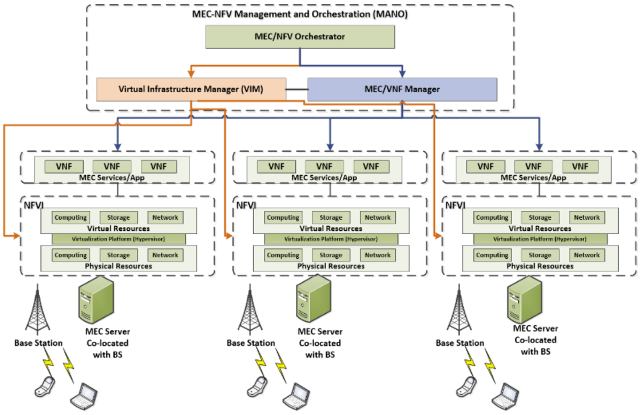

FIGURE 4. MEC-NFV Architectural Framework.

VI. LEARNED LESSONS AND PERSPECTIVE

Having presented various related research works for designing and deploying MEC systems, we present the learned lessons and our perspective to implement an auto-scaled proactive MEC-NFV infrastructure from scratch at the mobile operator site. Our proposed framework ensures that the mobile operator can efficiently support and process the received heterogeneous computation requests from mobile users with a high acceptance rate, less delay, guaranteed QoS, and less cost. We rely on the ETSI MEC reference architecture in an NFV environment model shown in Figure 3 [122] to design the MEC-NFV architectural framework. In this model, ETSI proposed the following changes [122] in the MEC architecture: (1) MEC applications are deployed as VNFs and are managed and orchestrated by the NFV components (i.e., Virtual Infrastructure Manager "VIM," Virtual Network Function Manager "VNFM," and Network Function Virtualization Orchestrator); (2) VNFs are managed using VNFM on the MEC platform; (3) the VIM manages the virtualized computing and communication resources of MEC infrastructure, (4) MEC platform manager is responsible for assigning the VNFs lifecycle to the VNFM; (5) MEC application Orchestrator is linked to the NFVO for resources and services orchestration.

In our proposed MEC-NFV architectural framework, we simplified the ETSI MEC-NFV reference architecture in Figure 4 and selected the blocks and reference points required to deploy a scalable proactive system. This enhanced system is expected to handle the dynamic heavy computational traffic received from the mobile end-user and support real-time low latency applications and services with guaranteed QoS.

MEC-NFV Management and Orchestration block should proactively scale VNFs in synergy with varying network traffic dynamics using the machine learning model output. The VNFs are dynamically placed on the MEC nodes to offer low latency and high-quality services based on the auto-scaling decision.

We considered a geographical region with no MEC infrastructure and an expected increase in computation-intensive traffic in this framework. We also assumed that the MEC nodes are located within the radio access network at the base station.

VII. CONCLUSION

Mobile Edge Computing provides an alternative to the conventional centralized Cloud processing with improved QoE and low latency to mobile end-users applications. As the number of computation-intensive and critical-mission applications increases, MEC infrastructure implementations become essential. We have defined four phases to implement a MEC infrastructure and proposed the high-level steps to deploy combined end-to-end MEC-NFV infrastructure. We compared current design and sizing approaches that produce a blueprint of a MEC infrastructure to support critical-latency applications and heavy dynamic workloads. We identified the importance of virtualizing the MEC infrastructure using the NFV concept to provide scalable and flexible infrastructure regardless of the underlying physical hardware. We classified existing VNF placement and auto-scaling mechanisms that could be used in MEC-NFV infrastructure. We analyzed the main approaches of MEC frameworks. Then, we reviewed the optimization approaches for MEC infrastructure to meet the required QoS threshold. The optimization approaches were explored depending on the component that will benefit from the MEC environment. Finally, we proposed a high-level understanding of the necessary considerations of building a proactive auto-scaled MEC-NFV infrastructure for dynamic mobile traffic support.

REFERENCES

[1] Ericsson. (Nov. 2020). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf

[2] 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.

[3] S. Singh, "Optimize cloud computations using edge computing," in *Proc. Int. Conf. Big Data, IoT Data Sci. (BID)*, Dec. 2017, pp. 49–53.

[4] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao, "5G on the horizon: Key challenges for the radio-access network," *IEEE Veh. Technol. Mag.*, vol. 8, no. 3, pp. 47–53, Sep. 2013.

[5] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[6] Y. Zhao, W. Wang, Y. Li, C. C. Meixner, M. Tornatore, and J. Zhang, "Edge computing and networking: A survey on infrastructures and applications," *IEEE Access*, vol. 7, pp. 101213–101230, 2019.

[7] N. Hassan, K.-L. A. Yau, and C. Wu, "Edge computing in 5G: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019.

[8] M. Patel, Y. Hu, J. Joubert, and C. Thornton, "Mobile-edge computing introductory technical white paper," Mobile-Edge Comput. (MEC) Ind. Initiative, New York, NY, USA, White Paper, 2014, pp. 854–864.

[9] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The Internet of Things, fog and cloud continuum: Integration and challenges," *Internet Things*, vols. 3–4, pp. 134–155, Oct. 2018.

[10] 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 11, 2015.

[11] A. Takeda, T. Kimura, and K. Hirata, "Evaluation of edge cloud server placement for edge computing environments," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, May 2019, pp. 1–2.

[12] D. Lu, Y. Qu, F. Wu, H. Dai, C. Dong, and G. Chen, "Robust server placement for edge computing," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2020, pp. 285–294.

[13] J. Meng, C. Zeng, H. Tan, Z. Li, B. Li, and X.-Y. Li, "Joint heterogeneous server placement and application configuration in edge computing," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2019, pp. 488–497.

[14] Z. Cao, H. Zhang, and B. Liu, "Performance and stability of application placement in mobile edge computing system," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Nov. 2018, pp. 1–8.

[15] D. Harris, J. Naor, and D. Raz, "Latency aware placement in multi-access edge computing," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 132–140.

[16] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, "Optimized placement of scalable IoT services in edge computing," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 189–197.

[17] G. Fodor, J. Vinogradova, P. Hammarberg, K. K. Nagalapur, Q. Zhiqiang, H. Do, R. Blasco, and M. U. Baig, "5G new radio for automotive, rail, and air transport," 2021, *arXiv:2101.08874*.

[18] H. Haile, K. J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, "End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks," *Comput. Netw.*, vol. 186, pp. 1–22, Feb. 2020.

[19] S. Gangakhedkar, H. Cao, A. R. Ali, K. Ganesan, M. Gharba, and J. Eichinger, "Use cases, requirements and challenges of 5G communication for industrial automation," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[20] F.-C. Kuo, F. A. Zdarsky, J. Lessmann, and S. Schmid, "Cost-efficient wireless mobile backhaul topologies: An analytical study," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[21] V. G. Nguyen, A. Brunström, K. J. Grinnemo, and J. Taheri, "SDN helps velocity in big data," in *Big Data and Software Defined Networks*, 1st ed. London, U.K.: IET Digital Library, 2018, pp. 207–228.

[22] K. Antevski, C. J. Bernardos, L. Cominardi, A. de la Oliva, and A. Mourad, "On the integration of NFV and MEC technologies: Architecture analysis and benefits for edge robotics," *Comput. Netw.*, vol. 175, Jul. 2020, Art. no. 107274.

[23] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, and K. W. Wen, "MEC in 5G networks," ETSI, Sophia Antipolis, France, White Paper 28, 2018.

[24] 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.

[25] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018.

[26] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in *Proc. 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, Nov. 2015, pp. 1–8.

[27] Q. V. Pham, T. L. Anh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, vol. 6, pp. 75868–75885, 2018.

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

[29] N. W. Sung, N.-T. Pham, T. Huynh, and W.-J. Hwang, "Predictive association control for frequent handover avoidance in femtocell networks," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 924–927, May 2013.

[30] Y. Dong, Z. Chen, P. Fan, and K. B. Letaief, "Mobility-aware uplink interference model for 5G heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2231–2244, Mar. 2016.

[31] L. N. T. Huynh, Q.-V. Pham, X. Q. Pham, T. D. T. Nguyen, M. D. Hossain, and E. N. Huh, "Efficient computation offloading in multi-tier multi-access edge computing systems: A particle swarm optimization approach," *Appl. Sci.*, vol. 10, no. 1, pp. 1–17, 2019.

[32] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generat. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[33] R. R. Sarukkai and A. Mendhekar, "Method and apparatus for accessing targeted, personalized voice/audio web content through wireless devices," U.S. Patent 6 728 731, Apr. 27, 2004.

[34] G. Simmons, G. A. Armstrong, and M. G. Durkin, "An exploration of small business website optimization: Enablers, influencers and an assessment approach," *Int. Small Bus. J.*, vol. 29, no. 5, pp. 534–561, Feb. 2011.

[35] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Proc. IEEE 7th Int. Symp. Service-Oriented Syst. Eng.*, Mar. 2013, pp. 320–323.

[36] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Jun. 2010.

[37] R. R. Sarukkai and A. Mendhekar, "Method and apparatus for accessing targeted, personalized voice/audio web content through wireless devices," U.S. Patent 6 728 731, Apr. 27, 2004.

[38] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[39] D. Dev and R. Patgiri, "Dr. Hadoop: An infinite scalable metadata management for Hadoop—How the baby elephant becomes immortal," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 1, pp. 15–31, Jan. 2016.

[40] S. Madakam and R. Ramaswamy, "The state of art: Smart cities in India: A literature review report," *Int. J. Innov. Res. Develop.*, vol. 2, no. 12, pp. 115–119, 2013.

[41] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct./Dec. 2017.

[42] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2866–2880, Oct. 2016.

[43] H. Xiang, X. Xu, H. Zheng, S. Li, T. Wu, W. Dou, and S. Yu, "An adaptive cloudlet placement method for mobile applications over GPS big data," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[44] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019.

[45] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2018, pp. 66–73.

[46] F. Zeng, Y. Ren, X. Deng, and W. Li, "Cost-effective edge server placement in wireless metropolitan area networks," *Sensors*, vol. 19, no. 1, p. 32, 2019.

[47] X. Wang, M. Razo, M. Tacca, and A. Fumagalli, "Planning and online resource allocation for the multi-resource cloud infrastructure," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 2938–2943.

[48] D. Ta, S. Zhou, W. Cai, X. Tang, and R. Ayani, "Network-aware server placement for highly interactive distributed virtual environments," in *Proc. 12th IEEE/ACM Int. Symp. Distrib. Simulation Real-Time Appl.*, Oct. 2008, pp. 95–102.

[49] S. K. Kasi, M. K. Kasi, K. Ali, M. Raza, H. Afzal, A. Lasebae, B. Naeem, S. U. Islam, and J. J. P. C. Rodrigues, "Heuristic edge server placement in industrial Internet of Things and cellular networks," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10308–10317, Jul. 2021.

[50] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Feb. 2017.

[51] 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.

[52] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "SDMEC: Software defined system for mobile edge computing," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 88–93.

[53] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2016, pp. 1–6.

[54] I. Abdullahi, S. Arif, and S. Hassan, "Ubiquitous shift with information centric network caching using fog computing," in *Computational Intelligence in Information Systems*. Cham, Switzerland: Springer, 2015.

[55] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[56] 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.

[57] C. Aggarwal and K. Srivastava, "Securing IoT devices using SDN and edge computing," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Oct. 2016, pp. 877–882.

[58] N. Song, C. Gong, X. An, and Q. Zhan, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Commun.*, vol. 13, no. 3, pp. 156–164, 2016.

[59] N. Fernando, S. W. Loke, and W. Rahayu, "Computing with nearby mobile devices: A work sharing algorithm for mobile-edge-clouds," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 329–343, Apr. 2019.

[60] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.

[61] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[62] I. D. Cardoso, J. P. Barraca, C. Gonçalves, and R. L. Aguiar, "Seamless integration of cloud and fog networks," *Int. J. Netw. Manage.*, vol. 26, no. 6, pp. 435–460, 2016.

[63] 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.

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

[65] S. Dutta, T. Taleb, and A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[66] G. A. Carella, M. Pauls, L. Grebe, and T. Magedanz, "An extensible autoscaling engine (AE) for software-based network functions," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 219–225.

[67] M. M. Murthy, H. A. Sanjay, and J. Anand, "Threshold based auto scaling of virtual machines in cloud environment," *Intl. Conf. Netw. Parallel Comput.*, p. 247256, 2014.

[68] C. Hung, Y. Hu, and K. Li, "Auto-scaling model for computing system," *Intl. J. Hybrid Info. Tech.*, vol. 5, no. 2, pp. 181–186, Apr. 2012.

[69] C. H. T. Arteaga, F. Risso, and O. M. C. Rendon, "An adaptive scaling mechanism for managing performance variations in network functions virtualization: A case study in an NFV-based EPC," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–7.

[70] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of autoscaling techniques for elastic applications in cloud environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, Dec. 2014.

[71] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, 1st Quart., 2020.

[72] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[73] S. Waidande, "A literature survey on scaling approaches for VNF in NFV monitoring," *Int. Res. J. Eng. Technol.*, vol. 5, no. 12, pp. 1–4, 2018.

[74] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of autoscaling techniques for elastic applications in cloud environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, Dec. 2014.

[75] A. Bilal, T. Tarik, A. Vajda, and B. Miloud, "Dynamic cloud resource scheduling in virtualized 5G mobile systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[76] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "Topology-aware prediction of virtual network function resource requirements," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 1, pp. 106–120, Mar. 2017.

[77] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett, and G. Estrada, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, 2017.

[78] N. Yu, Q. Xie, Q. Wang, H. Du, H. Huang, and X. Jia, "Collaborative service placement for mobile edge computing applications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[79] N. Kiran, X. Liu, S. Wang, and C. Yin, "VNF placement and resource allocation in SDN/NFV-enabled MEC networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2020, pp. 1–6.

[80] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 514–522.

[81] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2433–2443, Nov. 2018.

[82] S. Jošilo and G. Dán, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 207–220, Jan. 2019.

[83] S. Josilo and G. Dan, "Computation offloading scheduling for periodic tasks in mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 667–680, Apr. 2020.

[84] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.

[85] S. Josilo and G. Dan, "Joint management of wireless and computing resources for computation offloading in mobile edge clouds," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1507–1520, Oct. 2021.

[86] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 171–177.

[87] B. Brik, P. A. Frangoudis, and A. Ksentini, "Service-oriented MEC applications placement in a federated edge cloud architecture," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[88] L. Zhao and J. Liu, "Optimal placement of virtual machines for supporting multiple applications in mobile edge networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6533–6545, Jul. 2018.

[89] R. Urgaonkar, R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Perform. Eval.*, vol. 91, pp. 205–228, Sep. 2015.

[90] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1459–1467.

[91] Y. Zhang, L. Jiao, J. Yan, and X. Lin, "Dynamic service placement for virtual reality group gaming on mobile edge cloudlets," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1881–1897, Aug. 2019.

[92] H. Badri, T. Bahreini, D. Grosu, and K. Yang, "Energy-aware application placement in mobile edge computing: A stochastic optimization approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 909–922, Apr. 2020.

[93] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: An online learning approach," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1468–1476.

[94] T. Bahreini and D. Grosu, "Efficient algorithms for multi-component application placement in mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Nov. 17, 2020, doi: 10.1109/TCC.2020.3038626.

[95] S. Wang, M. Zafer, and K. K. Leung, "Online placement of multi-component applications in edge computing environments," *IEEE Access*, vol. 5, pp. 2514–2533, 2017.

[96] M. Wang, B. Cheng, W. Feng, and J. Chen, "An efficient service function chain placement algorithm in a MEC-NFV environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[97] M. A. Khoshkholghi, J. Taheri, D. Bhamare, and A. Kassler, "Optimized service chain placement using genetic algorithm," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 472–479.

[98] I. Jang, D. Suh, S. Pack, and G. Dán, "Joint optimization of service function placement and flow distribution for service function chaining," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2532–2541, Nov. 2017.

[99] *Requirements*, Standard ETSI GS MEC 002 V2.1.1, Oct. 2018.

[100] 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.

[101] 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, pp. 1–14, Oct. 2017.

[102] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[103] 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.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] A. Leivadeas, M. Falkner, I. Lambadaris, M. Ibnkahla, and G. Kesidis, "Balancing delay and cost in virtual network function placement and chaining," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 433–440.

[109] 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.

[110] 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, Nov. 2018, Art. no. e3504.

[111] 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.

[112] 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.

[113] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8099–8110, Sep. 2020.

[114] 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.

[115] 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.

[116] 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, 2018.

[117] 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.

[118] 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.

[119] J. Wen, C. Ren, and A. K. Sangaiah, "Energy-efficient device-to-device edge computing network: An approach offloading both traffic and computation," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 96–102, Sep. 2018.

[120] 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.

[121] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui, and H. V. Poor, "A machine learning approach for task and resource allocation in mobile-edge computing-based networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1358–1372, Feb. 2021.

[122] "Multi-access edge computing (MEC); Framework and reference architecture," ETSI, Sophia Antipolis, France, White Paper 003 v2.2.1 Jan. 2019.

LINA A. HAIBEH received the M.S. degree in electrical telecommunication engineering from the Université du Québec en Abitibi-Témiscamingue, Canada, in 2018. She is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, University of Ottawa, Ottawa, ON, Canada.

She is interested in the resource management in large-scale distributed systems, software-defined networking and network function virtualization, network performance evaluation and optimization, and network intelligence. Her research interests include voice over IP, 5G networks, wireless networks, network virtualization, cloud computing, edge computing, and mobile ad-hoc networks.

MUSTAPHA C. E. YAGOUB (Senior Member, IEEE) received the Dipl.-Ing. degree in electronics and the Magister degree in telecommunications from École Nationale Polytechnique, Algiers, Algeria, in 1979 and 1987, respectively, and the Ph.D. degree from the Institute National Polytechnique, Toulouse, France, in 1994.

After few years working in industry as a design engineer, he joined the Institute of Electronics, Université des Sciences et de la Technologie Houari Boumédiene, Algiers, as a Lecturer, from 1983 to 1991, and then as an Assistant Professor, from 1994 to 1999, and the Head of the Communication Department, from 1996 to 1999. From 1999 to 2001, he was a Visiting Scholar with the Department of Electronics, Carleton University, Ottawa, ON, Canada, working on neural networks applications in microwave areas. In 2001, he joined the School of Electrical Engineering and Computer Science (EECS), University of Ottawa, Ottawa, ON, Canada, where he is currently a Professor. He has authored or coauthored over 500 publications in these topics in international journals and referred conferences. He has also authored *Conception De Circuits Linéaires Et Non Linéaires Micro-Ondes* (Cépadues, Toulouse, France, 2000). His research interests include wireless communications systems design, RF/microwave CAD, RFID design, antenna design, active device modeling and characterization, neural networks for high frequency applications, and applied electromagnetics.

Dr. Yagoub is a Senior Member of the IEEE Microwave Theory and Techniques Society and a member of the Professional Engineers of Ontario, Canada, and the Ordre des ingénieurs du Québec, Canada.

ABDALLAH JARRAY (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Montreal, QC, Canada, in 2005 and 2010, respectively.

His research interests include network virtualization, cloud computing, wireless body area networks, femtocell networks, and optical networks. He is also interested on the development of optimization techniques for network design problems, integer linear programming, decomposition approaches, column generation, metaheuristics, and game theory.