A Survey on Mobile Edge Computing: The Communication Perspective

Yuyi Mao, *Student Member, IEEE*, Changsheng You, *Student Member, IEEE*, Jun Zhang, *Senior Member, IEEE*, Kaibin Huang, *Senior Member, IEEE*, and Khaled B. Letaief, *Fellow, IEEE*

***Abstract*—Driven by the visions of Internet of Things and 5G communications, recent years have seen a paradigm shift in mobile computing, from the centralized mobile cloud computing toward mobile edge computing (MEC). The main feature of MEC is to push mobile computing, network control and storage to the network edges (e.g., base stations and access points) so as to enable computation-intensive and latency-critical applications at the resource-limited mobile devices. MEC promises dramatic reduction in latency and mobile energy consumption, tackling the key challenges for materializing 5G vision. The promised gains of MEC have motivated extensive efforts in both academia and industry on developing the technology. A main thrust of MEC research is to seamlessly merge the two disciplines of wireless communications and mobile computing, resulting in a wide-range of new designs ranging from techniques for computation offloading to network architectures. This paper provides a comprehensive survey of the state-of-the-art MEC research with a focus on joint radio-and-computational resource management. We also discuss a set of issues, challenges, and future research directions for MEC research, including MEC system deployment, cache-enabled MEC, mobility management for MEC, green MEC, as well as privacy-aware MEC. Advancements in these directions will facilitate the transformation of MEC from theory to practice. Finally, we introduce recent standardization efforts on MEC as well as some typical MEC application scenarios.**

***Index Terms*—Mobile edge computing, fog computing, mobile cloud computing, computation offloading, resource management, green computing.**

I. Introduction

THE LAST decade has seen Cloud Computing emerging as a new paradigm of computing. Its vision is the centralization of computing, storage and network management in the Clouds, referring to data centers, backbone IP networks and cellular core networks [1], [2]. The vast resources available in the Clouds can then be leveraged to deliver elastic computing power and storage to support resource-constrained end-user devices. Cloud Computing has been driving the rapid growth of many Internet companies. For example, the Cloud business has risen to be the most profitable sector for Amazon [3], and Dropbox's success depended highly on the Cloud service of Amazon.

However, in recent years, a new trend in computing is happening with the function of Clouds being increasingly moving towards the network edges [4]. It is estimated that tens of billions of Edge devices will be deployed in the near future, and their processor speeds are growing exponentially, following Moore's Law. Harvesting the vast amount of the idle computation power and storage space distributed at the network edges can yield sufficient capacities for performing computation-intensive and latency-critical tasks at mobile devices. This paradigm is called *Mobile Edge Computing* (MEC) [5]. While long propagation delays remain a key drawback for Cloud Computing, MEC, with the proximate access, is widely agreed to be a key technology for realizing various visions for next-generation Internet, such as Tactile Internet (with millisecond-scale reaction time) [6], *Internet of Things* (IoT) [7], and Internet of Me [8]. Presently, researchers from both academia and industry have been actively promoting MEC technology by pursuing the fusion of techniques and theories from both disciplines of *mobile computing* and *wireless communications*. This paper aims at providing a survey of key research progress in this young field from the communication perspective. We shall also present a research outlook containing an ensemble of promising research directions for MEC.

A. Mobile Computing for 5G: From Clouds to Edges

In the past decade, the popularity of mobile devices and the exponential growth of mobile Internet traffic have been driving the tremendous advancements in wireless communications and networking. In particular, the breakthroughs in small-cell networks, multi-antenna, and millimeter-wave communications promise to provide users gigabit wireless access in next-generation systems [9]. The high-rate and highly-reliable air interface allows to run computing services of mobile devices at the remote cloud data center, resulting in the research area called *Mobile Cloud Computing* (MCC). However, there is an inherent limitation of MCC, namely, the long propagation distance from the end user to the remote cloud center, which will result in excessively long

Manuscript received January 4, 2017; revised June 12, 2017; accepted August 15, 2017. Date of publication August 25, 2017; date of current version November 21, 2017. This work was supported by the Hong Kong Research Grants Council under Grant Nos. 16200214, 17209917, and 17259416. *(Corresponding author: Changsheng You.)*

Y. Mao and J. Zhang are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: ymaoac@connect.ust.hk; eejzhang@ust.hk).

C. You and K. Huang are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: csyou@eee.hku.hk; huangkb@eee.hku.hk).

K. B. Letaief is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, and also with Hamad bin Khalifa University, Doha, Qatar (e-mail: eekhaled@ust.hk).

Digital Object Identifier 10.1109/COMST.2017.2745201

latency for mobile applications. MCC is thus not adequate for a wide-range of emerging mobile applications that are latency-critical. Presently, new network architectures are being designed to better integrate the concept of Cloud Computing into mobile networks, as will be discussed in the latter part of this article.

In 5G wireless systems, ultra-dense edge devices, including small-cell *base stations* (BSs), wireless *access points* (APs), laptops, tablets, and smartphones, will be deployed, each having a computation capacity comparable with that of a computer server a decade ago. As such, a large population of devices will be idle at every time instant. It will, in particular, be harvesting enormous computation and storage resources available at the network edges, which will be sufficient to enable ubiquitous mobile computing. In a nutshell, the main target of wireless systems, from 1G to 4G, is the pursuit of increasingly higher wireless speeds to support the transition from voice-centric to multimedia-centric traffic. As wireless speeds approach the wireline counterparts, the mission of 5G is different and much more complex, namely to support the explosive evolution of ICT and Internet. In terms of functions, 5G systems will support *communications, computing, control and content delivery* (4C). In terms of applications, a wide-range of new applications and services for 5G are emerging, such as real-time online gaming, *virtual reality* (VR) and *ultra-high-definition* (UHD) video streaming, which require unprecedented high access speed and low latency. The past decade also saw the take-off of different visions of next-generation Internet including IoT, Tactile Internet (with millisecond latency), Internet-of-Me, and social networks. In particular, it was predicted by Cisco that about 50 billion IoT devices (e.g., sensors and wearable devices) will be added to the Internet by 2020, most of which have limited resources for computing, communication and storage, and have to rely on Clouds or edge devices for enhancing their capabilities [10]. It is now widely agreed that relying only on Cloud Computing is inadequate to realize the ambitious millisecond-scale latency for computing and communication in 5G. Furthermore, the data exchange between end users and remote Clouds will allow the data tsunami to saturate and bring down the backhaul networks. This makes it essential to supplement Cloud Computing with MEC that pushes traffic, computing and network functions towards the network edges. This is also aligned with a key characteristic of next-generation networks that information is increasingly *generated locally and consumed locally*, which arises from the booming of applications in IoT, social networks and content delivery [4].

The concept of MEC was firstly proposed by the *European Telecommunications Standard Institute* (ETSI) in 2014, and was defined as a new platform that "*provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers*" [5]. The original definition of MEC refers to the use of BSs for offloading computation tasks from mobile devices. Recently, the concept of *Fog Computing* has been proposed by Cisco as a generalized form of MEC where the definition of edge devices gets broader, ranging from smartphones to set-top boxes [11]. This led to the emergence of a new research area called Fog Computing and Networking [4], [12], [13]. However, the areas of Fog Computing and MEC are overlapping and the terminologies are frequently used interchangeably. In this paper, we focus on MEC but many technologies discussed are also applicable to Fog Computing.

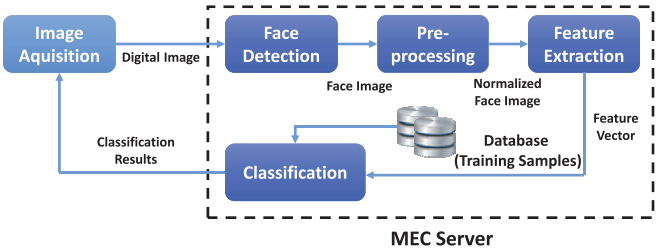


Fig. 1. Main computation components in a face recognition application [17].

MEC is implemented based on a virtualized platform that leverages recent advancements in *network functions virtualization* (NFV), *information-centric networks* (ICN) and *software-defined networks* (SDN). Specifically, NFV enables a single edge device to provide computing services to multiple mobile devices by creating multiple *virtual machines* (VMs)[1] for simultaneously performing different tasks or operating different network functions [15]. On the other hand, ICN provides an alternative end-to-end service recognition paradigm for MEC, shifting from a host-centric to an information-centric one for implementing context-aware computing. Last, SDN allows MEC network administrators to manage services via function abstraction, achieving scalable and dynamic computing [16]. A main focus of MEC research is to develop these general network technologies so that they can be implemented at the network edges.

There is an increasing number of emerging mobile applications that will benefit from MEC, by offloading their computation-intensive tasks to the MEC servers for cloud execution. In the following, we will provide two examples to illustrate the basic principles of MEC. One is the face recognition application as shown in Fig. 1, which typically consists of five main computation components, including image acquisition, face detection, pre-processing, feature extraction, and classification [17]. While the image acquisition component needs to be executed at the mobile device for supporting the user interface, the other components could be offloaded for cloud processing, which contain complex computation such as signal processing and *machine learning* (ML) algorithms. Another popular stream of applications that can leverage the rich resources at the network edges are *augmented reality* (AR) applications, which are able to combine the computer-generated data with physical reality. AR applications as shown in Fig. 2 have five critical components [18], [19], namely, the video source (which obtains raw video frames from the mobile camera), a tracker (which tracks the position of the user), a mapper (which builds a model of the environment), an object recognizer (which identifies known objects in the

[1]The VM is a virtual computer mapped to the physical machine's hardwares, providing virtual CPU, memory, hard drive, network interface, and other devices [14].

TABLE I
COMPARISON OF MEC AND MCC SYSTEMS

	MEC	MCC
Server hardware	Small-scale data centers with moderate resources [5], [20]	Large-scale data centers (each contains a large number of highly-capable servers) [21], [22]
Server location	Co-located with wireless gateways, WiFi routers, and LTE BSs [5]	Installed at dedicated buildings, with size of several football fields [23], [24]
Deployment	Densely deployed by telecom operators, MEC vendors, enterprises, and home users. Require lightweight configuration and planning [5]	Deployed by IT companies, e.g., Google and Amazon, at a few locations over the world. Require sophisticated configuration and planning [21]
Distance to end users	Small (tens to hundreds of meters) [15]	Large (may across continents) [25]
Backhaul usage	Infrequent use Alleviate congestion [26]	Frequent use Likely to cause congestion [26]
System management	Hierarchical control (centralized/distributed) [27]	Centralized control [27]
Supportable latency	Less than tens of milliseconds [15], [28]	Larger than 100 milliseconds [29], [30]
Applications	Latency-critical and computation-intensive applications, e.g., AR, automatic driving, and interactive online gaming [5], [31].	Latency-tolerant and computation-intensive applications, e.g., online social networking, and mobile commerce/health/learning [32]–[35].

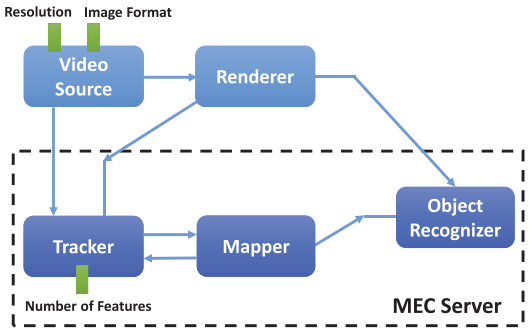


Fig. 2. Main computation components in an AR application [18].

environment), and a renderer (which prepares the processed frame for display). Among these components, the video source and renderer should be executed locally, while the most computation-intensive components, i.e., the tracker, mapper and object recognizer, can be offloaded for cloud execution. In this way, mobile users can enjoy various benefits from MEC such as latency reduction and energy savings, as will be elaborated in the next subsection.

B. Mobile Edge Computing Versus Mobile Cloud Computing

As shown in Table I, there exist significant disparities between MEC and MCC systems in terms of computing server, distance to end users and typical latency, etc. Compared with MCC, MEC has the advantages of achieving lower latency, saving energy for mobile devices, supporting context-aware computing, and enhancing privacy and security for mobile applications. These advantages are briefly described through some examples and applications in the following.

Low Latency: The latency for a mobile service is the aggregation of three components: *propagation*, *computation*, and *communication* latency, depending on the propagation distance, computation capacity, and data rate, respectively. First, the information-propagation distances for MEC are typically tens of meters for the cases of dense small-cell networks or *device-to-device* (D2D) transmissions, and typically no longer than 1km for general cases. In contrast, Cloud Computing requires transmissions from end users to nodes in core networks or data centers with distances ranging from tens of kilometers to that across continents. This results in much shorter propagation delay for MEC than that for MCC. Second, MCC requires the information to pass through several networks including the radio access network, backhaul network and Internet, where traffic control, routing and other network-management operations can contribute to excessive delay. With the communication constrained at the network edges, MEC is free from these issues. Last, for the computation latency, a Cloud has a massive computation power that is several orders of magnitude higher than that of an edge device (e.g., a BS). However, the Cloud has to be shared by a much larger number of users than an edge device, reducing their gap in the computation latency. Furthermore, a modern BS is powerful enough for running highly sophisticated computing programs. For instance, the edge cloud at a BS has 10^2-10^4 times higher computation capability than the minimum requirement (e.g., a CPU over 3.3GHz, 8GB RAM, 70GB storage space) for running the Call-of-Duty 13, a popular shooter game.[2] In general, experiments have shown that the total latency for MCC is in the range of 30-100ms [30]. This is unacceptable for many latency-critical mobile applications such as real-time online gaming, virtual sports and autonomous driving, which may require tactile speed with latency approaching 1ms [36]. In contrast, with short propagation distances and simple protocols, MEC has the potential of realizing tactile-level latency for latency-critical 5G applications.

Mobile Energy Savings: Due to their compact forms, IoT devices have limited energy storage but are expected to cooperate and perform sophisticated tasks such as surveillance, crowd-sensing and health monitoring [37]. Powering the tens of billions of IoT devices remains a key challenge for designing IoT given that frequent battery recharging/replacement is impractical if not impossible. By effectively

[2]https://www.callofduty.com/

supporting *computation offloading*, MEC stands out as a promising solution for prolonging battery lives of IoT devices. Specifically, computation-intensive tasks can be offloaded from IoT devices to edge devices so as to reduce their energy consumption. Significant energy savings by computation offloading have been demonstrated in experiments, e.g., the completion of up to 44-time more computation load for a multimedia application *eyeDentify* [38] or the increase of battery life by 30-50% for different AR applications [39].

Context-Awareness: Another key feature that differentiates MEC from MCC is the ability of an MEC server for leveraging the proximity of edge devices to end users to track their real-time information such as behaviors, locations, and environments. Inference based on such information allows the delivery of context-aware services to end users [40]–[42]. For instance, for the museum video guide, an AR application, can predict users' interests based on their locations in the museum to automatically deliver contents related to, e.g., artworks and antiques [43]. Another example is the CTrack system that uses the BS fingerprints to track and predict the trajectories of a large number of users for the purposes of traffic monitoring, navigation and routing, and personalized trip management [44].

Privacy/Security Enhancement: The capability of enhancing the privacy and security of mobile applications is also an attractive benefit brought by MEC compared to MCC. In MCC systems, the Cloud Computing platforms are the remote public large data centers, such as the Amazon EC2 and Microsoft Azure, which are susceptible to attacks due to their high concentration of information resources of users. In addition, the ownership and management of users' data are separated in MCC, which shall cause the issues of private data leakage and loss [45]. The use of proximate edge servers provides a promising solution to circumvent these problems. On one hand, due to the distributed deployment, small-scale nature, and the less concentration of valuable information, MEC servers are much less likely to become the target of a security attack. Second, many MEC servers could be private-owned cloudlets, which shall ease the concern of information leakage. Applications that require sensitive information exchange between end users and servers would benefit from MEC. For instance, the enterprise deployment of MEC could help avoid uploading restricted data and material to remote data centers, as the enterprise administrator itself manages the authorization, access control, and classifies different levels of service requests without the need of an external unit [46].

C. Paper Motivation and Outline

MEC has emerged as a key enabling technology for realizing the IoT and 5G visions [15], [47], [48]. MEC research lies at the intersection of mobile computing and wireless communications, where the existence of many research opportunities has resulted in a highly active area. In recent years, researchers from both academia and industry have investigated a wide-range of issues related to MEC, including system and network modeling, optimal control, multiuser resource allocation, implementation and standardization. Subsequently, several survey articles have been published to provide overviews of the MEC area with different focuses, including system models, architectures, enabling techniques, applications, edge caching, edge computation offloading, and connections with IoT and 5G [26], [27], [49]–[55]. Their themes are summarized as follows. An overview of MEC platforms is presented in [49] where different existing MEC frameworks, architectures, and their application scenarios, including FemtoClouds, REPLISM, and ME-VOLTE, are discussed. The survey of [50] focuses on the enabling techniques in MEC such as cloud computing, VM, NFV, SDN that allow the flexible control and multi-tenancy support. Liu *et al.* [51] categorize diverse MEC applications, service models, deployment scenarios, as well as network architectures. The survey in [52] presents a taxonomy for MEC applications and identifies potential directions for research and development, such as content scaling, local connectivity, augmentation, and data aggregation and analytics. In [27], emerging techniques of edge *computing, caching, and communications* (3C) in MEC are surveyed, showing the convergence of 3C. Besides, key enablers of MEC such as cloud technology, SDN/NFV, and smart devices are also discussed. The survey in [53] focuses on three critical design problems in computation offloading for MEC, namely, the offloading decision, computational resource allocation, and mobility management. In addition, the role of MEC in IoT, i.e., creating new IoT services, is highlighted in [54] through MEC deployment examples with reference to IoT use cases. Several attractive use scenarios of MEC in 5G networks are also introduced in [26], ranging from mobile-edge orchestration, collaborative caching and processing, and multi-layer interference cancellation. Furthermore, potential business opportunities related to MEC are discussed in [55] from the perspectives of application developers, service providers, and network equipment vendors. In view of prior work, there still lacks a systematic survey article providing comprehensive and concrete discussions on specific MEC research results with a deep integration of mobile computing and wireless communications, which motivates the current work. This paper differs from existing surveys on MEC in the following aspects. First, the current survey summarizes existing models of computing and communications in MEC to facilitate theoretical analysis and provide a quick reference for both researchers and practitioners. Next, we present a comprehensive literature review on joint radio-and-computational resource allocation for MEC, which is the central theme of the current paper. The literature review in our paper shall be a valuable addition to the existing survey literature on MEC, which can benefit readers from the research community in building up a systematic understanding of the state-of-the-art resource management techniques for MEC systems. Furthermore, we identify and discuss several research challenges and opportunities in MEC from the communication perspective, for which potential solutions are elaborated. In addition, to bridge the gap between theoretical research and real implementation of MEC, recent standardization efforts and use scenarios of MEC will then be introduced.

This paper is organized as follows. In Section II, we summarize the basic MEC models, comprising models of

TABLE II
SUMMARY OF IMPORTANT ACRONYMS

Acronym	Definition	Acronym	Definition
AF	application function	MEC	mobile edge computing
AR	augmented reality	ML	machine learning
AP	access point	mMTC	massive machine-type communication
BS	base station	NEF	network exposure function
CAPEX	capital expenditure	NFC	near-filed communications
C-RAN	cloud radio access network	NFV	network functions virtualization
CSI	channel-state information	OFDMA	orthogonal frequency-division multiple access
DAG	directed acyclic graph	PCF	policy control function
DCN	data-center network	PMR	peak-to-mean ratio
DNS	domain name system	PoC	proof of concept
DP	dynamic programming	QoS	quality of service
DPP	determinantal point process	RAM	random access memory
DVFS	dynamic frequency and voltage scaling	RAN	radio access network
D2D	device-to-device	RFID	radio frequency identification
EH	energy harvesting	RNIS	radio network information services
eMBB	enhanced mobile broadband	SDN	software-defined networks
ESI	energy side information	SINR	signal-to-interference-plus-noise ratio
ETSI	European Telecommunications Standard Institute	TOF	traffic offloading function
GLB	geographical load balancing	UE	user equipment
Het-MEC	heterogeneous MEC	UHD	ultra-high-definition
HetNets	heterogeneous networks	UPF	user plane function
HPPP	homogeneous Poisson point process	UPS	uninterrupted power supply
IaaS	Infrastructure as a Service	URLLC	ultra-reliable and low latency communication
ICN	information-centric networks	VM	virtual machine
ISG	industry specification group	VR	virtual reality
ISI	inter-symbol interference	V2X	vehicular-to-everything
IoT	Internet of Things	WPT	wireless power transfer
KKT	Karush-Kuhn-Tucker	3C	computing, caching, and communications
LP	linear programming	3GPP	3rd Generation Partnership Project
LTE	long-term evolution	4C	communications, computing, control and content delivery
MCC	mobile cloud computing	5GPPP	European 5G Infrastructure Public Private Partnership
MDP	Markov decision process	5QI	5G QoS Indicator

computation tasks, communications, mobile devices and MEC servers, based on which the models of MEC latency and energy consumption are developed. Next, a comprehensive review is presented in Section III, focusing on the research of joint radio-and-computational resource management for different types of MEC systems, including single-user, multiuser systems as well as multi-server MEC. Subsequently, a set of key research issues and future directions are discussed in Section IV including 1) deployment of MEC systems, 2) cache-enabled MEC, 3) mobility management for MEC, 4) green MEC, and 5) security-and-privacy issues in MEC. Specifically, we analyze the design challenges for each research problem and provide several potential research approaches. Last, the MEC standardization efforts and applications are reviewed and discussed in Section V, followed by concluding remarks in Section VI. We summarize the definitions of the acronyms that will be frequently used in this paper in TABLE II for ease of reference.

II. MEC COMPUTATION AND COMMUNICATION MODELS

In this section, system models are introduced for the key computation/communication components of the typical MEC system. The models provide mechanisms for abstracting various functions and operations into optimization problems and facilitating theoretical analysis as discussed in the following sections.

For the MEC system shown in Fig. 3, the key components include mobile devices (a.k.a. end users, clients, service subscribers) and MEC servers. The MEC servers are typically small-scale data centers deployed by the cloud and telecom operators in close proximity with end users and can be co-located with wireless APs. Through a gateway, the servers are connected to the data centers via Internet. Mobile devices and servers are separated by the air interface where reliable wireless links can be established using advanced wireless communication and networking technologies. In the following subsections, we will introduce the models for different components of MEC systems, including models for the computation tasks, wireless communication channels and networks, as well as the computation latency and energy consumption models of mobile devices and MEC servers.

A. Computation Task Models

There are various parameters that play critical roles in modeling the computation tasks, including latency, bandwidth utilization, context awareness, generality, and scalability [22]. Though it is highly sophisticated to develop accurate models for tasks, there exist simple ones that are reasonable and allow mathematical tractability. In this subsection, we introduce two computation-task models popularly used in existing literature on MCC and MEC, corresponding to binary and partial computation offloading, respectively.

1) Task Model for Binary Offloading: A highly integrated or relatively simple task cannot be partitioned and has to be

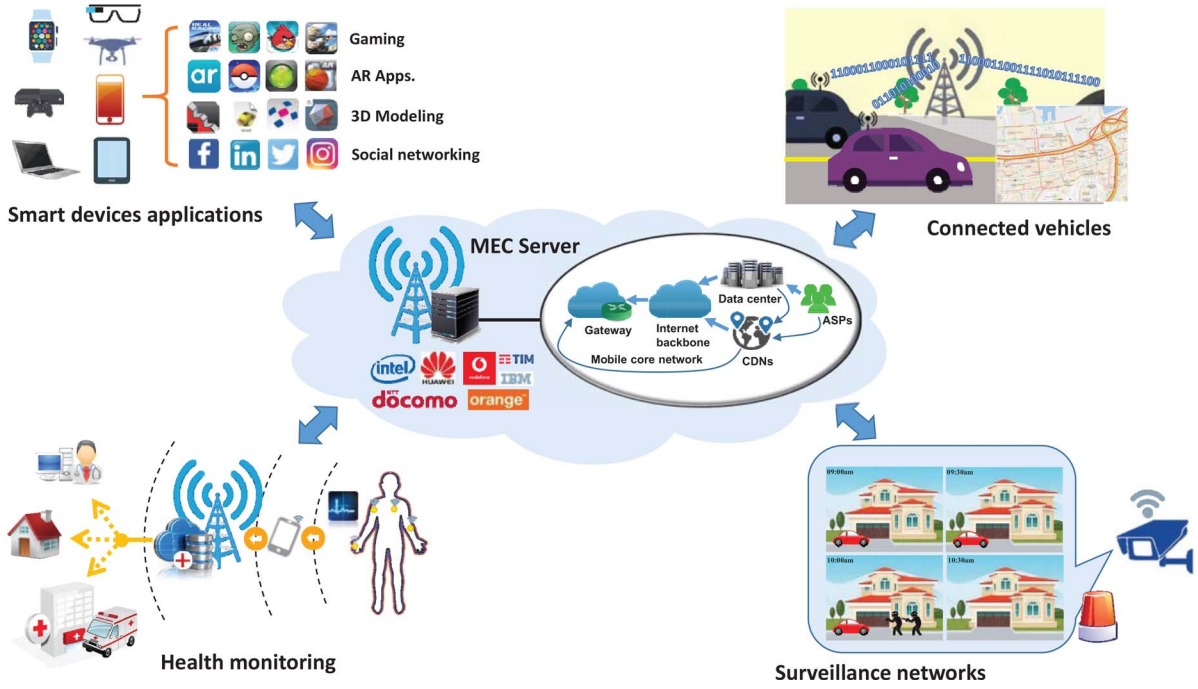


Fig. 3. Architecture of the MEC systems.

executed as a whole either locally at the mobile device or offloaded to the MEC server, called *binary offloading*. Such a task can be represented by a three-field notation $A(L, \tau_d, X)$. This commonly-used notation contains the information of the task input-data size L (in bits), the completion deadline τ_d (in second), and the computation workload/intensity X (in CPU cycles per bit). These parameters are related to the nature of the applications and can be estimated through task profilers [56], [57]. The use of these three parameters not only captures essential properties of mobile applications such as the computation and communication demands, but also facilitates simple evaluation of the execution latency and energy consumption performance (which will be analyzed in Section II-C).

The task $A(L, \tau_d, X)$ is required to be completed before a hard deadline τ_d. This model can also be generalized to handle the soft deadline requirement which allows a small portion of tasks to be completed after τ_d [58]. In this case, the number of CPU cycles needed to execute 1-bit of task input data is modeled as a random variable X. Specifically, define x_0 as a positive integer such that $\Pr(X > x_0) \leq \rho$ where ρ is a small real number: $0 < \rho \ll 1$. It follows that $\Pr(LX > W_\rho) \leq \rho$ where $W_\rho = Lx_0$. Then given the L-bit task-input data, W_ρ upper bounds the number of required CPU cycles almost surely.

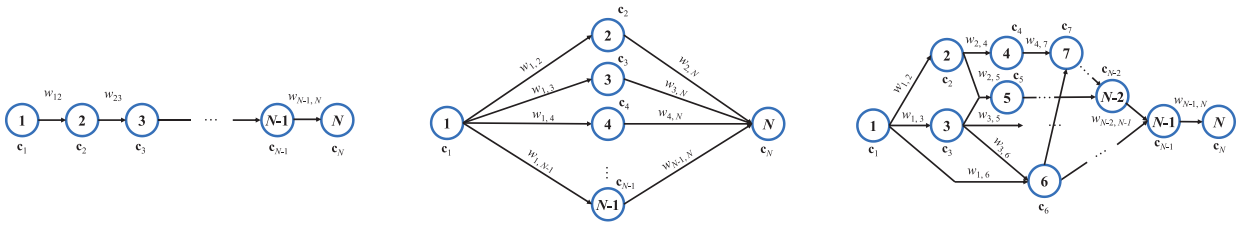
2) Task Models for Partial Offloading: In practice, many mobile applications are composed of multiple procedures/components (e.g., the computation components in an AR application as shown in Fig. 2), making it possible to implement fine-grained (partial) computation offloading. Specifically, the program can be partitioned into two parts with one executed at the mobile device and the other offloaded for edge execution.

The simplest task model for partial offloading is the *data-partition model*, where the task-input bits are bit-wise independent and can be arbitrarily divided into different groups and executed by different entities in MEC systems, e.g., parallel execution at the mobiles and MEC server.

Nevertheless, the dependency among different procedures/components in many applications cannot be ignored as it significantly affects the procedure of execution and computation offloading due to the following reasons:

- First, the execution order of functions or routines cannot be arbitrarily chosen because the outputs of some components are the inputs of others.
- Second, due to either software or hardware constraints, some functions or routines can be offloaded to the server for remote execution, while the ones can only be executed locally such as the image display function.

This calls for task models that are more sophisticated than the mentioned data-partition model that can capture the inter-dependency among different computation functions and routines in an application. One such model is called the *task-call graph*. The graph is typically a *directed acyclic graph* (DAG), which is a finite directed graph with no directed cycles. We shall denote it as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the set of vertices $\mathcal{V}$ represents different procedures in the application and the set of edges $\mathcal{E}$ specifies their call dependencies. There are three typical dependency models of sub-tasks (i.e., task components such as functions or routines), namely *sequential*, *parallel*, and *general* dependency [59], [60], as illustrated in Fig. 4. For the mobile initiated applications, the first and the last steps, e.g., collecting the I/O data and displaying the computation results on the screen, are normally required to be executed locally. Thus, node 1 and node N in Fig. 4(a)–4(c) are components that must be executed locally. Besides, the required computation workloads and resources of each procedure, e.g., the number of required CPU cycles and the amount of needed memory, can also be specified in the vertices of the task-call graph,



(a) Sequential dependency (b) Parallel dependency (c) General dependency

Fig. 4. Typical topologies of the task-call graphs.

while the amount of input/output data of each procedure can be characterized by imposing weights on the edges.

B. Communication Models

In the literature of MCC, communication channels between the mobile devices and cloud servers are typically abstracted as bit pipes with either constant rates or random rates with given distributions. Such coarse models are adopted for tractability and may be reasonable for the design of MCC systems where the focuses are to tackle the latency in the core networks and management of large-scale cloud but not the wireless-communication latency. The scenario is different for MEC systems. Given small-scale edge clouds and targeting latency-critical applications, reducing communication latency by designing a highly efficient air interface is the main design focus. Consequently, the mentioned bit-pipe models are insufficient as they overlook some fundamental properties of wireless propagation and are too simplified to allow the implementation of advanced communication techniques. To be specific, wireless channels differ from the wired counterparts in the following key aspects [61]:

1) Due to atmospheric ducting, reflection and refraction from scattering objects in the environment (e.g., buildings, walls and trees), there exists the well-known *multi-path fading* in wireless channels, making the channels highly time-varying and can cause severe *inter-symbol interference* (ISI). Thus, effective ISI suppression techniques, such as equalization and spread spectrum, are needed for reliable transmissions.
2) The broadcast nature of wireless transmissions results in a signal being interfered by other signals occupying the same spectrum, which reduces their respective receive *signal-to-interference-plus-noise ratios* (SINRs) and thereby results in the probabilities of error in detection. To cope with the performance degradation, interference management becomes one of the most important design issues for wireless communication systems and has attracted extensive research efforts [62]–[64].
3) Spectrum shortage has been the main foe for very high-rate radio access, motivating extensive research on exploiting new spectrum resources [65], [66], designing novel transceiver architectures [67]–[69] and network paradigms [70], [71] to improve the spectrum efficiency, as well as developing spectrum sharing and aggregation techniques to facilitate efficient use of fragmented and underutilized spectrum resources [72]–[74].

The random variations of wireless channels in time, frequency and space make it important for designing efficient MEC systems to seamlessly integrate control of computation offloading and radio resource management. For instance, when the wireless channel is in deep fade, the reduction on execution latency by remote execution may not be sufficient to compensate for the increase of transmission latency due to the steep drop in transmission-data rates. For such cases, it is desirable to defer offloading till the channel gain is favorable or switch to an alternative frequency/spatial channel with a better quality for offloading. Furthermore, increasing transmission power can increase the data rate, but also lead to a larger transmission energy consumption. The above considerations necessitate the joint design of offloading and wireless transmissions, which should be adaptive to the time-varying channels based on the accurate *channel-state information* (CSI).

In MEC systems, communications are typically between APs and mobile devices with the possibility of direct D2D communications. The MEC servers are small-scale data centers deployed by the Cloud Computing/telecom operators, which can be co-located with the wireless APs, e.g., the public WiFi routers and BSs, as so to reduce the *capital expenditure* (CAPEX) (e.g., site rental). As shown in Fig. 3, the wireless APs not only provide the wireless interface for the MEC servers, but also enable the access to the remote data center through backhaul links, which could help the MEC server to further offload some computation tasks to other MEC servers or to large-scale cloud data centers. For the mobile devices that cannot communicate with MEC servers directly due to insufficient wireless interfaces, D2D communications with neighboring devices provide the opportunity to forward the computation tasks to MEC servers. Furthermore, D2D communications also enable the peer-to-peer cooperation on resource sharing and computation-load balancing within a cluster of mobile devices.

Presently, there exist different types of commercialized technologies for mobile communications, including the *near-filed communications* (NFC), *radio frequency identification* (RFID), Bluetooth, WiFi, and cellular technologies such as the *long-term evolution* (LTE). Besides, the 5G network, which will be realized by the development of LTE in combination with new radio-access technologies, is currently being standardized and will be put into commercial use as early as 2020 [75]. These technologies can support wireless offloading from mobiles to

TABLE III
CHARACTERISTICS OF TYPICAL WIRELESS COMMUNICATION TECHNOLOGIES

	NFC	RFID	Bluetooth	WiFi	LTE	5G
Max. Coverage	10cm	3m	100m	100m	up to 5km	Excellent coverage
Operation Freq.	13.56MHz	LF: 120-134kHz HF: 13.56MHz UHF: 850-960MHz	2.4GHz	2.4GHz, 5GHz	TDD: 1.85-3.8GHz FDD: 0.7-2.6GHz	6-100GHz
Data Rate	106, 212, 414kbps	Low (LF) to high (UHF)	22Mbps	135Mbps (IEEE 802.11n)	DL: 300Mbps UL: 75Mbps	Indoor/dense outdoor: up to 10Gbps Urban/suburban: > hundreds of Mbps

APs or peer-to-peer mobile cooperation for varying data rates and transmission ranges. We list the key characteristics of typical wireless communication technologies in Table III, which differ significantly in terms of the operation frequency, maximum coverage range, and data rate. For NFC, the coverage range and data rate are very low and thus the technology is suitable for applications that require little information exchange, e.g., *e*-payment and physical access authentication. RFID is similar to NFC, but only allows one-way communications. Bluetooth is a more powerful technique to enable short-range D2D communications in MEC systems. For long-range communications between mobiles and MEC servers, WiFi and LTE (or 5G in the future) are two primary technologies enabling the access to MEC systems, which can be adaptively switched depending on their link reliability. For the deployment of wireless technologies in MEC systems, the communication and networking protocols need to be redesigned to integrate both the computing and communication infrastructures, and effectively improve the computation efficiency that is more sophisticated than the data transmission.

C. Computation Models of Mobile Devices

In this subsection, we introduce the computation models of mobile devices and discuss methodologies of evaluating the computation performance.

The CPU of a mobile device is the primary engine for local computation. The CPU performance is controlled by the CPU-cycle frequency f_m (also known as the CPU clock speed). The state-of-the-art mobile CPU architecture adopts the advanced *dynamic frequency and voltage scaling* (DVFS) technique, which allows stepping-up or -down of the CPU-cycle frequency (or voltage), resulting in growing and reducing energy consumption, respectively. In practice, the value of f_m is bounded by a maximum value, $f_{\text{CPU}}^{\max}$, which reflects the limitation of the mobile's computation capability. Based on the computation task model introduced in Section II-A, the execution latency for task $A(L, \tau, X)$ can be calculated accordingly to

$$t_m = \frac{LX}{f_m}, \tag{1}$$

which indicates that a high CPU clock speed is desirable in order to reduce the execution latency, at the cost of higher CPU energy consumption.

As the mobile devices are energy-constrained, the energy consumption for local computation is another critical measurement for the mobile computing efficiency. According to the circuit theory [76]–[79], the CPU power consumption can be divided into several factors including the *dynamic*, *short-circuit*, and *leakage* power consumption,[3] where the dynamic power consumption dominates the others. In particular, it is shown in [78] that the dynamic power consumption is proportional to the product of $V_{\text{cir}}^2 f_m$ where V_{cir} is the circuit supplied voltage. It is further noticed in [76] and [79] that, the clock frequency of the CPU chip is approximately linear proportional to the voltage supply when operating at the low voltage limits. Thus, the energy consumption of a CPU cycle is given by κf_m^2, where κ is a constant related to the hardware architecture. For the computation task $A(L, \tau, X)$ with CPU clock speed f_m, the energy consumption can be derived:

$$E_m = \kappa L X f_m^2. \tag{2}$$

One can observe from (1) and (2) that the mobile device may not be able to complete a computation-intensive task within the required deadline, or else the energy consumption incurred by mobile execution is so high that the onboard battery will be depleted quickly. In such cases, offloading the task execution process to an MEC server is desirable.

Besides CPUs, other hardware components in the mobile devices, e.g., the *random access memory* (RAM) and flash memory, also contribute to the computation latency and energy consumption [80], while detailed discussions are beyond the scope of this survey.

D. Computation Models of MEC Servers

In this subsection, we introduce the computation models of the MEC servers. Similar as the mobile devices, the computation latency and energy consumption are of particular interests.

The server-computation latency is *negligible* compared with communication or local-computation latency in MEC systems where the computation loads for servers are much lower than their computation capacities [79], [81]. This model can be also relevant for multiuser MEC systems with resource-constrained servers if the servers' computation loads are regulated by multiuser resource management under latency and computation-capacity constraints [82].

[3]The dynamic power consumption comes from the toggling activities of the logic gates inside a CPU, which shall charge/discharge the capacitors inside the logic gates. When a logic gate toggles, some of its transistors may change states, and thus, there might be a short period of time when some transistors are conducting simultaneously. In this case, the direct path between the source and ground will result in some short-circuit power loss. The leakage power dissipation is due to the flowing current between doped parts of the transistors [78], available on https://en.wikipedia.org/wiki/CPU_power_dissipation.

On the other hand, as edge servers have relatively limited computational resources, it is necessary to consider the *non-negligible* server execution time in the general design of MEC systems, yielding the computation model for the severs discussed in the remainder of this subsection. Two possible models are considered in the literature, corresponding to the *deterministic* and *stochastic* server-computation latency. The deterministic model is proposed to consider the exact server-computation latency for latency-sensitive applications, which is implemented using techniques such as VMs and DVFS. Specifically, assume the MEC server allocates different VMs for different mobile devices, allowing independent computation [83]. Let $f_{s,k}$ denote the allocated servers' CPU-cycle frequency for mobile device k. Similar to Section II-C, it follows that the server execution time denoted by $t_{s,k}$ can be calculated as $t_{s,k} = \frac{w_k}{f_{s,k}}$, where w_k is the number of required CPU cycles for processing the offloaded computation workload. This model has been widely used for designing computation-resource allocation policies [84]–[86]. A similar model was proposed in [82], where the MEC server is assumed to perform load balancing for the total offloaded computation workloads. In other words, the CPU cycles at the MEC server are proportionally allocated to each mobile device such that they experience the same execution latency. Furthermore, in addition to the CPU processing time, the server scheduling queuing delay should be accounted for MEC servers with relatively small computation capacities, where parallel computing via virtualization techniques is not feasible and thus it needs to process the computation workloads sequentially. Without loss of generality, denote k as the processing order for a mobile device and name it as mobile k. Thus, the total server-computation latency including the queuing delay for device k denoted by $T_{s,k}$ can be given as

$$T_{s,k} = \sum_{i \le k} t_{s,i}. \tag{3}$$

For latency-tolerant applications, the average server-computation time can be derived based on stochastic models. For example, in [87], the task arrivals and service time are modeled by the Poisson and exponential processes, respectively. Thus, the average server-computation time can be derived using techniques from queuing theory. Last, for all above models, as investigated in [1], multiple VMs sharing the same physical machine will introduce the I/O interference among different VMs. It results in the longer computation latency for each VM denoted by $T'_{s,k}$, which can be modeled by $T'_{s,k} = T_{s,k}(1+\epsilon)^n$ where ϵ is the performance degradation factor as the percentage increasing of the latency [88].

The energy consumption of an MEC server is jointly determined by the usage of the CPU, storage, memory, and network interfaces. Since the CPU contribution is dominant among these factors, it is the main focus in the literature. Two tractable models are widely used for the energy consumption of MEC servers. One model is based on the DVFS technique described as follows. Consider an MEC server that handles K computation tasks and the k-th task is allocated with w_k CPU cycles with CPU-cycle frequency $f_{s,k}$. Hence, the total energy consumed by the CPU at the MEC server, denoted by E_s, can be expressed as

$$E_s = \sum_{k=1}^{K} \kappa w_k f_{s,k}^2, \tag{4}$$

which is similar to that for the mobile devices. The other model is based on an observation in recent works [89]–[91] that the server-energy consumption is *linear* to the CPU utilization ratio which depends on the computation load. Moreover, even for an idle server, it still, on average, consumes up to 70% of the energy consumption for the case with the full CPU speed. Thus, the energy consumption at the MEC server can be calculated according to

$$E_s = \alpha E_{\max} + (1-\alpha) E_{\max} u, \tag{5}$$

where $E_{\max}$ is the energy consumption for a fully-utilized server, α is the fraction of the idle energy consumption (e.g., 70%) and u denotes the CPU utilization ratio. This model suggests that energy-efficient MEC should allow servers to be switched into the sleep mode in the case of light load and consolidation of computation loads into fewer active servers.

E. Summary and Insights

The MEC computation and communication models are summarized in Fig. 5, laying the foundation for the analysis of MEC resource management in the next section. These models shed several useful insights on the offloading design, listed as follows.

- The effective design of MEC should leverage and integrate advanced techniques from both areas of wireless communications and mobile computing.
- It is vital to choose suitable computation task models for different MEC applications. For example, the soft-deadline task model can be applied for social networking applications but is not suitable for AR applications due to the stringent computation latency requirements. Moreover, for a specific application, the task model also depends on the offloading scenario, e.g., the data-partition model can be used when the input-data is offloaded, and the task-call graph should be considered when each task component can be offloaded as a whole.
- The wireless channel condition significantly affects the amount of energy consumption for computation offloading. MEC has the potential to reduce the transmission energy consumption due to short distances between users and MEC servers. Advanced wireless communication techniques, such as interference cancelation and adaptive power control, can further reduce the offloading energy consumption.
- Dynamic CPU-cycle frequency control is the key technique for controlling the computation latency and energy consumption for both mobile devices and MEC servers. Specifically, increasing the CPU-cycle frequency can reduce the computing time but contributes to higher energy consumption. The effective CPU-cycle frequency control should approach the optimal tradeoff between computation latency and energy consumption.

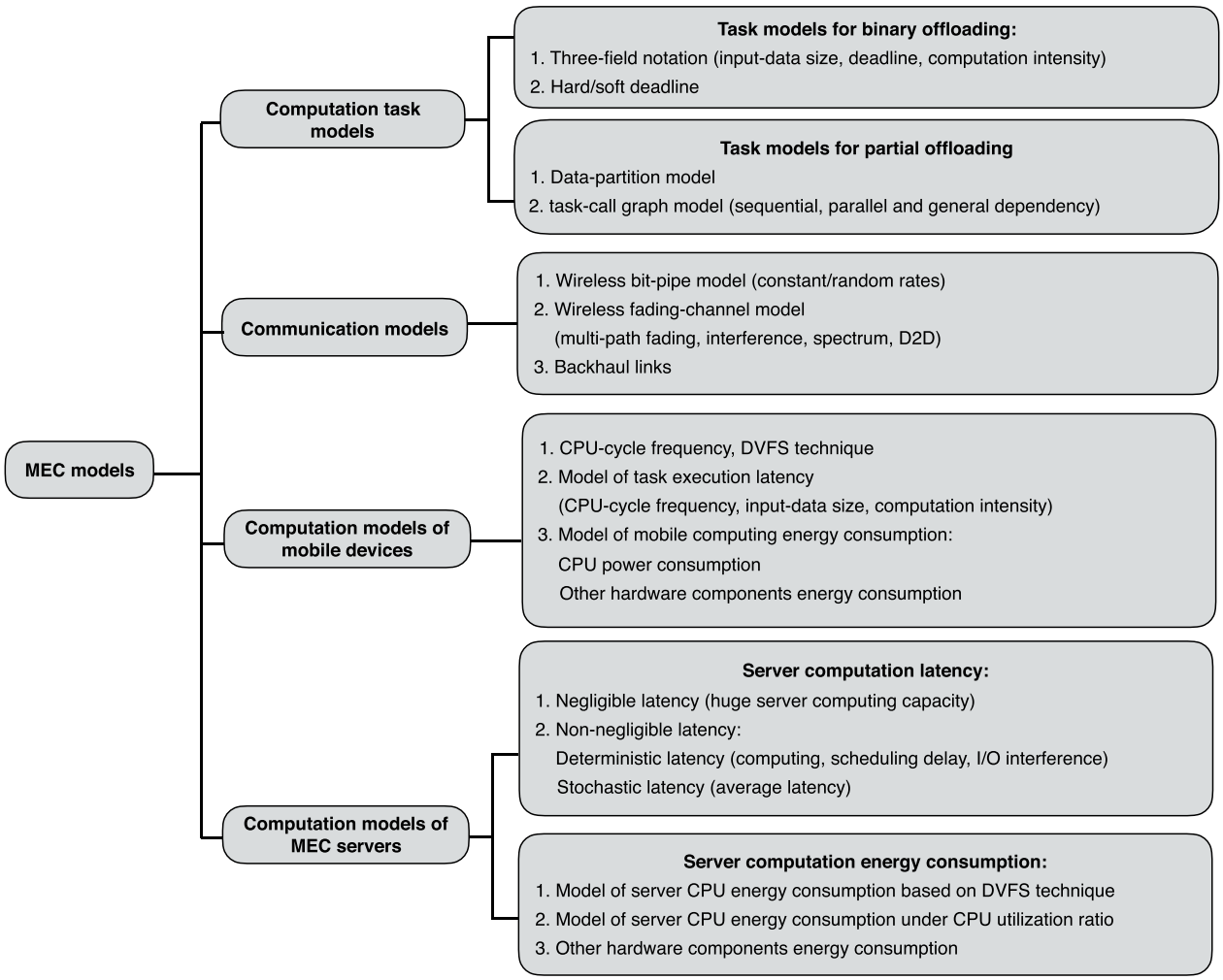


Fig. 5. Summary of MEC models.

- Apart from the task-execution latency, the computation scheduling delay is non-negligible if the MEC server has a relatively small computation capacity or heavy computation loads are offloaded to the server. Load-balancing and intelligent scheduling policies can be designed to reduce the total computation latency.

III. Resource Management in MEC Systems

The joint radio-and-computational resource management plays a pivotal role in realizing energy-efficient and low-latency MEC. The implementation of relevant techniques is facilitated by the network architecture where MEC servers and wireless APs (e.g., BSs and WiFi routers) are co-located. In this section, we provide a comprehensive overview of the literature on resource management for MEC systems summarized in Fig. 6. Our discussion starts from the simple single-user systems comprising a single mobile device and a single MEC server, allowing the exposition of the key design considerations and basic design methodologies. Subsequently, more complex multiuser MEC systems are considered where multiple offloading users compete for the use of both the radio and server-computational resources and have been coordinated. Last, we extend the discussion to MEC systems with heterogeneous servers which not only provide the freedom of server selection but also allow the cooperation among servers. Such network-level operations can significantly enhance the performance of MEC systems.

A. Single-User MEC Systems

This subsection focuses on the simple single-user MEC systems and reviews a set of recent research efforts for this case. The discussion is divided according to three popularly-used task models, namely, deterministic task model with binary offloading, deterministic task model with partial offloading, and stochastic task model.

1) Deterministic Task Model With Binary Offloading: Consider the mentioned single-user MEC system where the binary offloading decision is on whether a particular task should be offloaded for edge execution or local computation. The investigations for the optimal offloading policies can be dated back to those for conventional Cloud Computing systems, where the communication links were typically assumed to have a fixed rate B. In [92] and [93], general guidelines were developed for determining the offloading decision for the purposes of minimizing the mobile-energy consumption and computation latency. Denote w as the amount

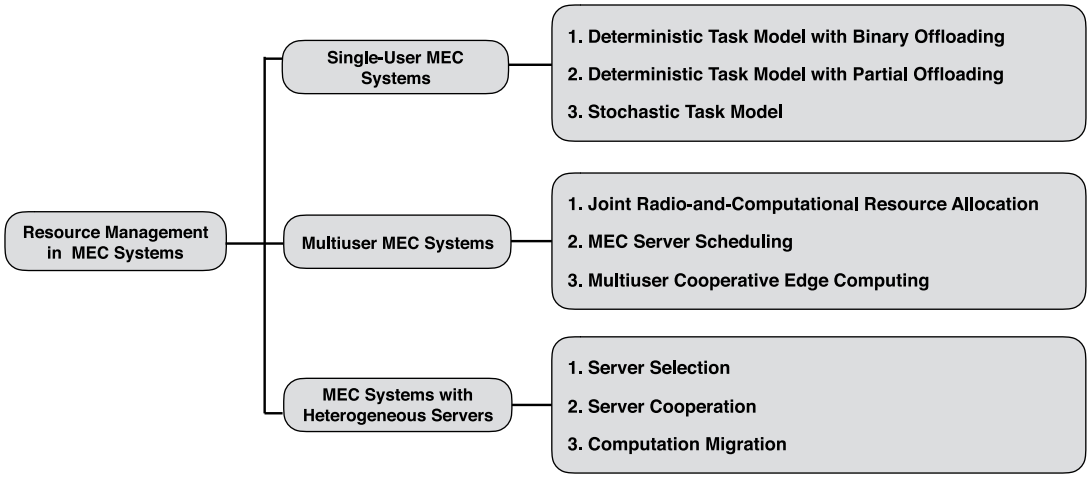


Fig. 6. Classification of resource management techniques for MEC.

of computation (in CPU cycles) for a task, f_m as the CPU speed of the mobile device, d as the input data size, and f_s as the CPU speed at the cloud server. Offloading the computation to the cloud server can improve the latency performance only when

$$\frac{w}{f_m} > \frac{d}{B} + \frac{w}{f_s}, \tag{6}$$

which holds for applications that require heavy computation and have small amount of data input, or when the cloud server is fast, and the transmission rate is sufficiently high. Moreover, let p_m represent the CPU power consumption at the mobile device, and p_t as the transmission power, p_i as the power consumption at the device when the task is running at the server. Offloading the task could help save mobile energy when

$$p_m \times \frac{w}{f_m} > p_t \times \frac{d}{B} + p_i \times \frac{w}{f_s} \tag{7}$$

holds, i.e., applications with heavy computation and light communication should be offloaded.

Nevertheless, the data rates for wireless communications are not constant and change with the time-varying channel gains as well as depend on the transmission power. This calls for the design of control policies for power adaptation and data scheduling to streamline the offloading process. In addition, as the CPU power consumption increases super-linearly with the CPU-cycle frequency, the computation energy consumption for mobile execution can be minimized using DVFS techniques. These issues led to the active field of adaptive MEC as summarized below.

In [94], the problem of transmission-energy minimization under a computation-deadline constraint was formulated with the optimization variable being the input-data transmission time, where the famous Shannon-Hartley formula gives the power-rate function. The optimization problem is convex and can be solved in closed form. In particular, task offloading is desirable when the channel power gain is greater than a threshold and the server CPU is fast enough, which reveals the effects of wireless channels on the offloading decision. A further study was conducted by Zhang *et al.* [79] to minimize the energy consumption for executing a task with a soft real-time requirement, targeting, e.g., multimedia applications, which requires the task to be completed within the deadline with a given probability ρ. The offloading decision was determined by the computation mode (either offloading or local computing) that incurs less energy consumption. On one hand, the energy consumption for local execution was optimized using the DVFS technique, which was formulated as a convex optimization problem with the objective function being the expected energy consumption of the W_ρ CPU cycles and a time duration constraint for these CPU cycles. The optimal CPU-cycle frequencies over the computation duration were derived in closed form by solving the *Karush-Kuhn-Tucker* (KKT) conditions, suggesting that the processor should speed up as the number of completed CPU cycles increases. On the other hand, the expected energy consumption for task offloading was minimized via data transmission scheduling. Under the Gilbert-Elliott channel model, the optimal data transmission scheduling was obtained through *dynamic programming* (DP) techniques, and the scaling law of the minimum expected energy consumption with respect to the execution deadline was also derived. This framework was further developed in [81] where both the local computing and offloading are powered by wireless energy transfer. Specifically, the optimal CPU-cycle frequencies for local computing and time division for offloading should be adaptive to the transferred power.

2) Deterministic Task Model With Partial Offloading: The running of a relatively sophisticated mobile application can be decomposed into a set of smaller sub-tasks. Inspired by recent advancements of parallel computing, partial offloading (also known as program partitioning) schemes were proposed to further optimize MEC performance in [59], [60] and [95]–[100].

In [95], full granularity in program partitioning was considered where the task-input data can be arbitrarily divided for local and remote executions. Joint optimization of the offloading ratio, transmission power and CPU-cycle frequency was performed to minimize the mobile-energy consumption (or latency) subject to a latency (or energy consumption) constraint. Both the energy and latency minimization problems are non-convex in contrast to the ones for binary-offloading. The former problem can be solved optimally with a

variable-substitution technique while a sub-optimal algorithm was proposed for the latter one in [95].

In [59], [60], and [96]–[100], applications were modeled by task-call graphs discussed earlier that specify the dependency among different sub-tasks, and the code partitioning schemes designed to dynamically generate the optimal set of tasks for offloading. In [59], by leveraging the concept of load balancing between the mobile device and the server, a heuristic program-partitioning algorithm was developed to minimize the execution latency. Kao *et al.* investigated the latency minimization problem with a prescribed resource utilization constraint in [96], and proposed a polynomial-time approximate solution with guaranteed performance. To maximize the energy savings achieved by computation offloading, the scheduling and cloud offloading decisions were jointly optimized using an integer programming approach in [60]. In [97], considering the wireless channel models including the block fading channel, *independent and identical distributed* (i.i.d.) stochastic channel, and the Markovian stochastic channel, the expected energy consumption minimization problem with a completion time constraint was found to be a *stochastic shortest-path* problem, and the *one-climb* policies (i.e., the execution only migrates once from the mobile device to the server) were shown to be optimal. In addition, the program-partitioning schemes were also optimized together with the physical layer parameters, such as the transmission and reception power, constellation size, as well as the data allocation for different radio interfaces [98]–[100].

3) Stochastic Task Model: Resource management policies have also been developed for MEC systems with stochastic task models characterized by random task arrivals, where the arrived but not yet executed tasks join the queues in task buffers [101]–[106]. For such systems, the long-term performance, e.g., the long-term average energy consumption and execution latency, are more relevant compared with those of deterministic task arrivals, and the temporal correlation of the optimal system operations makes the design more challenging. As a result, the design of MEC systems with random task arrivals is an area less explored compared with the simpler cases with deterministic task models. In [101], in order to minimize the mobile-energy consumption while keeping the proportion of executions violating the deadline requirement below a threshold, a dynamic offloading algorithm was proposed to determine the offloaded software components from an application running at a mobile user based on Lyapunov optimization techniques, where 3G and WiFi networks are accessible to the device but their rates vary at different locations. Assuming that concurrent local and edge executions are feasible, the latency-optimal task scheduling policies were designed in [102] based on the theory of *Markov decision process* (MDP), which controls the states of the local processing and transmission units and the task buffer queue length based on the channel state. It was shown that the optimal task-scheduling policy significantly outperforms the greedy scheduling policy (i.e., tasks are scheduled to the local CPU/transmission unit whenever they are idle). To jointly optimize the computation latency and energy consumption, the problem of minimizing the long-term average execution cost was considered in [100] and [104], where the former only optimized the offloading data size based on the MDP theory while the latter jointly controlled the local CPU frequency, modulation scheme as well as data rates under a semi-MDP framework. In [105], the energy-latency tradeoff in MEC systems with heterogeneous types of applications was investigated, including the non-offloadable workload, cloud-offloadable workload and network traffic. A Lyapunov optimization-based algorithm was proposed to jointly decide the offloading policy, task allocation, CPU clock speed, and selected network interface. It was also shown that the energy consumption decreases inversely proportional to V while the latency increases linearly with V, where V is a control parameter in the proposed algorithm. Similar investigation was conducted for MEC systems with a multi-core mobile device in [106].

4) Summary and Insight: The comparison of resource management schemes for single-user MEC systems is shown in Table IV. This series of work yields a number of useful insights on controlling computation offloading as summarized below.

- Consider binary offloading. For energy savings, computation offloading is preferred to local computation when the user has desirable channel condition or small local computation capability. Moreover, beamforming and MIMO techniques can be exploited to reduce the energy consumption for offloading. For latency reduction, computation offloading is advantageous over local computation when the user has a large bandwidth and the MEC server is provisioned with huge computation capacity.
- Partial offloading allows flexible components/data partitioning. By offloading time-consuming or energy-consuming sub-tasks to MEC servers, partial offloading can achieve larger energy savings and smaller computation latency compared with binary offloading. Graph theory is a powerful tool for designing the offloading scheduling according to the task dependency graph.
- For stochastic task models, the temporal correlation of task arrivals and channels can be exploited to design adaptive dynamic computation offloading policies. Moreover, it is critical to maintain the task buffer stability at the user and MEC server via offloading rate control.

B. Multiuser MEC Systems

While the preceding subsection aims at resource management policies for single-user MEC systems with a dedicated MEC server, this subsection considers the multiuser MEC systems comprising multiple mobile devices that share one edge server. Several new challenges are investigated in the sequel, including the multiuser joint radio-and-computational resource allocation, MEC server scheduling, and multiuser cooperative edge computing.

1) Joint Radio-and-Computational Resource Allocation: Compared with the central cloud, the MEC servers have much less computational resources. Therefore, one key issue in designing a multiuser MEC system is how to allocate the finite radio-and-computational resources to multiple mobiles

TABLE IV
THE COMPARISON OF PAPERS FOCUSING ON SINGLE-USER MEC SYSTEMS

Task model	Design Objective	Reference	Proposed Solution
Binary offloading	Energy	[79]	Optimize local computing and offloading by controlling the CPU frequency and transmission rate
		[81]	Propose a novel framework of wirelessly powered MEC and optimize both local computing and offloading
		[92]	Propose general guidelines to make offloading decision for energy consumption minimization
		[94]	Propose the optimal binary computation offloading decision using convex optimization
	Energy and latency	[93]	Propose general guidelines to make offloading decision for energy-consumption and computation-latency minimization
Partial offloading	Energy	[60]	Propose a joint scheduling and computation offloading algorithm by parallel processing appropriate components in the mobile and cloud
		[97]	Formulate a stochastic shortest-path problem and derive the one-climb optimal policy
		[99]	Jointly optimize the program partitioning with the selection of transmission power and constellation size
		[100]	Propose an iterative algorithm for the optimal offloading scheduling and the percentage of the data to be carried on each radio interface
	Latency	[59]	Propose a heuristic load-balancing program-partitioning algorithm
		[96]	Propose a polynomial-time approximate solution with guaranteed performance
	Energy and latency	[95]	Jointly optimize the offloading ratio, transmission power and CPU-cycle frequency using variable-substitution technique
		[98]	Propose an algorithmic to leverage the structure of the call graphs by means of message passing under both serial and parallel implementations of processing and communication
Stochastic model	Energy	[101]	Propose a Lyapunov optimization-based dynamic computation offloading policy
	Latency	[102]	Dynamically control the local processing and transmission using MDP
		[103]	Optimize local computing and transmission using semi-MDP and propose a one-dimensional heuristic search algorithm
	Energy and Latency	[104]	Jointly control the local CPU frequency, modulation scheme as well as the data rates under a semi-MDP framework
		[105]	Propose a Lyapunov optimization-based algorithm to decide the offloading policy, task allocation, CPU clock speed, and selected network interface
		[106]	Propose a Lyapunov optimization-based scheme for cloud offloading scheduling, as well as download scheduling for cloud execution output

for achieving a system-level objective, e.g., the minimum sum mobile-energy consumption. Both the centralized and distributed resource allocation schemes have been studied for different MEC systems as reviewed in the following.

For centralized resource allocation [82], [84], [99], [107]–[112], the MEC server obtains all the mobile information, including the CSI and computation requests, makes the resource-allocation decisions, and informs the mobile devices about the decisions. In [82], mobile users time-share a single edge server and have different computation workloads and local-computation capacities. A convex optimization problem was formulated to minimize the sum mobile-energy consumption. The key finding is that the optimal policy for controlling offloading data size and time allocation has a simple threshold-based structure. Specifically, an offloading priority function was firstly derived according to mobile users' channel conditions and local computing energy consumption. Then, the users with priorities above and below a given threshold will perform full and minimum offloading (so as to meet a given computation deadline), respectively. This result was also extended to the OFDMA-based MEC systems for designing a close-to-optimal computation offloading policy. In [84], instead of controlling the offloading data size and time, the MEC server determined the mobile-transmission power and assigned server CPU cycles to different users in order to reduce the sum mobile-energy consumption. The optimal solution shows that, there exists an optimal one-to-one mapping between the transmission power and the number of allocated CPU cycles for each mobile device. This work was further extended in [99] to account for the optimal binary offloading based on the model of task-call graphs. Ren *et al.* [110] considered the multiuser video compression offloading in MEC and minimized the latency in local compression, edge cloud compression and partial compression offloading scenarios. Besides, in order to minimize the energy and delay cost for multiuser MEC systems where each user has multiple tasks, Chen *et al.* jointly optimized the offloading decisions and the allocation of communication resource via a *separable semidefinite relaxation* approach in [111], which was later extended in [112] by taking the computational resource allocation and processing cost into account. Different from [82], [84], [99], [110]–[112], the revenue of service providers was maximized in [107] under constraints of *quality of service* (QoS) requirements for all mobile devices. The assumed fixed resource usage of each user results in a semi-MDP problem, which was transformed into a *linear programming* (LP) model and efficiently solved. In [108], assuming a stochastic task arrival model, the energy-latency tradeoff in multiuser MEC systems was investigated via a Lyapunov optimization-based online algorithm, which jointly manages

the available radio-and-computational resources. Centralized resource management for multiuser MEC system based on *cloud radio access network* (C-RAN) has also been investigated in [109].

Another thrust of research targets distributed resource allocation for multiuser MEC systems which were designed using game theory and decomposition techniques [85], [86], [113]–[117]. In [85] and [113], the computation tasks were assumed to be either locally executed or fully offloaded via single and multiple interference channels, respectively. With fixed mobile-transmission power, an integer optimization problem was formulated to minimize the total energy consumption and offloading latency, which was proved to be NP-hard. Instead of designing a centralized solution, the game-theoretic techniques were applied to develop a distributed algorithm that is able to achieve a Nash equilibrium. Moreover, it was shown that for each user, offloading is beneficial only when the received interference power is lower than a threshold. Furthermore, this work was extended in [114] and [115], where each mobile has multiple tasks and can offload computation to multiple APs connected by a common edge-server, respectively. For the offloading process, in addition to the transmission energy, this work has also accounted for the scanning energy of the APs and the fixed circuit power. The proposed distributed offloading policy shows that a mobile device should handover the computation to a different AP only when a new user choosing the same AP achieves a larger benefit. Building on the system model in [85], the joint optimization for the mobile-transmission power and the CPU-cycle allocation of the edge server was investigated in [86]. To solve the formulated mixed-integer problem, the decomposition technique was utilized to optimize the resource allocation and offloading decision sequentially. Specifically, the offloading decision problem was reduced to a sub-modular maximization problem and solved by designing a heuristic greedy algorithm. Similar decomposition technique and successive convex approximation technique were utilized in [116] and [117] respectively to design distributed resource allocation algorithms for MEC systems.

2) MEC Server Scheduling: The works discussed earlier [82], [84]–[86], [107], [115] are based on the assumptions of user synchronization and the feasibility of parallel local-and-edge computation. However, studying practical MEC server scheduling requires relaxation of these assumptions as discussed below together with the resultant designs. First, the arrival times of different users are in general asynchronous so that it is desirable for the edge server with finite computational resource to buffer and compute the tasks sequentially, which incurs the queuing delay. In [118], to cope with the bursty task arrivals, the server scheduling was integrated with uplink-downlink transmission scheduling to minimize the average latency using queuing theory. Second, even for synchronized task arrivals, the latency requirements can differ significantly over users running different types of applications ranging from latency-sensitive to latency-tolerant applications. This fact calls for the server scheduling to assign users different levels of priorities based on their latency requirements. In [119], after the pre-resource allocation, the MEC server will check the deadline of different tasks during the server computing process and adaptively adjust the task execution order to satisfy the heterogeneous latency requirements. Last, some computation tasks each consists of several dependent sub-tasks such that the scheduling of these modules must satisfy the task-dependency requirements. The task model with a sequential sub-task arrangement was considered in [120] that jointly optimizes the program partitioning for multiple users and the server-computation scheduling to minimize the average completion time. As a result, a heuristic algorithm was proposed to solve the formulated mixed-integer problem. Specifically, it first optimizes the computation partition for each user. Under these partitions, it will search the time intervals violating the resource constraint and adjust them accordingly. Furthermore, the general dependency-task model as shown in Fig. 4(c) was considered for multiple users in [116]. This model drastically complicates the computing time characterization. To address this challenge, a measure of *ready time* was defined for each sub-task as the earliest time when all the predecessors have been computed. Then, the offloading decision, mobile CPU-cycle frequency and mobile-transmission power were jointly optimized to reduce the sum mobile-energy consumption and computation latency with a proposed distributed algorithm.

3) Multiuser Cooperative Edge Computing: Multiuser cooperative computing is envisioned as a promising technique to improve the MEC performance by providing two advantages [121]–[127]. First, MEC servers with limited computational resources may be overloaded when they have to serve a large number of offloading mobile users. In such cases, the burdens on the servers can be lightened via peer-to-peer mobile cooperative computing. Second, sharing the computational resources among the users can balance the uneven distribution of the computation workloads and computation capabilities over users. In [121], D2D communication was proposed to enable multiuser cooperative computing. In particular, this work studied how to detect and utilize computational resources on other users. This idea was adopted in [122] to propose a D2D-based heterogeneous MCC networks. Such a novel framework was shown to enhance the network capacity and offloading probability. Moreover, for wireless sensor networks, cooperative computing was proposed in [123] to enhance its computation capability. First, the optimal computation partition for minimizing the total energy consumption of two cooperative nodes was investigated. This result was then utilized to design the fairness-aware energy-efficient cooperative node selection. Furthermore, Song *et al.* [124] showed that sharing computation results among the peer users can significantly reduce the communication traffic for a multiuser MEC system. Assuming the task can either be offloaded or computed locally, a mixed-integer optimization problem was formulated to minimize the total energy consumption under the constraint of the system communication traffic. To tackle this challenging problem, two online task scheduling algorithms were proposed based on pricing and Lyapunov optimization theories, respectively. In addition, by employing a helper, a four-slot joint computation-and-communication cooperation protocol was proposed in [125], where the helper not only computes part of the tasks offloaded from the user, but also acts as a

TABLE V
THE COMPARISON OF PAPERS FOCUSING ON MULTIUSER MEC SYSTEMS

Theme	Design Type/Motivation	Design Objective	Reference	Proposed Solution
Joint radio-and-computational resource allocation	Centralized	Energy	[82]	Design the optimal threshold-based resource allocation policy based on defined offloading priority functions for TDMA and OFDMA systems
			[84]	Jointly optimize the allocation of communication and computational resources
			[99]	Design the optimal resource allocation and code partitioning by call-graph selection approach
			[109]	Solve the non-convex resource allocation problem for C-RAN using iterative algorithms
		Latency	[110]	Minimize the latency in multiuser video compression via resource allocation
		Energy and latency	[108]	Propose a Lyapunov optimization-based dynamic computation offloading policy
			[111], [112]	Jointly optimize the offloading decisions and the allocation of resource via semidefinite relaxation
		Revenue	[107]	Design the optimal resource allocation based on semi-MDP
	Distributed	Energy	[117]	Propose a distributed iterative algorithm using successive convex approximation techniques
		Energy and latency	[85], [113]	Develop a distributed algorithm that is able to achieve the Nash equilibrium
			[114]	Propose a distributed algorithm for multiuser MEC systems where each user has multiple tasks
			[115]	Consider multiple servers and develop a distributed algorithm admitting the Nash equilibrium
			[116]	Propose a decomposition algorithm to control the computation offloading selection, clock frequency control and transmission power allocation iteratively
		Utility	[86]	Propose a decomposition algorithm to optimize the resource allocation and offloading decisions
MEC server scheduling	Bursty data arrivals	Latency	[118]	Optimize the uplink and downlink scheduling using queuing theory
	Heterogeneous deadlines	Energy	[119]	Propose a pre-resource allocation and joint scheduling scheme
	Task dependency	Latency	[120]	Propose heuristic algorithms with searching and adjusting phases based on constraint relaxation
		Energy and latency	[116]	Propose a decomposition algorithm to control the computation offloading selection, clock frequency control and transmission power allocation iteratively
Cooperative computing	D2D communication	Task success rate	[121]	Propose the optimal and periodic mobile cloud access scheme
		Network capacity and offloading probability	[122]	Propose D2D communication techniques in heterogeneous MEC systems
	Cooperation	Energy	[123]	Propose a fairness-aware energy-efficient cooperative node selection scheme
			[125]	Propose a four-slot protocol to enable joint computation and communication cooperation
	Share computation results	Energy	[124]	Propose a Lyapunov optimization-based cooperative computing policy
	Share computational resource	Energy	[126]	Propose a "string-pulling" offloading policy based on constructed offloading feasibility tunnel
	Small BSs cooperation	Delay cost	[127]	Propose a peer offloading framework that allows both centralized and autonomous decision making

relay node to forward the tasks to the MEC server. Another recent work [126] investigated the optimal offloading policies in a peer-to-peer cooperative computing system where the computing helper has time-varying computational resources. Specifically, an *offloading feasibility tunnel* was constructed based on the helper's CPU profile and buffer size. Given the tunnel, the optimal offloading was shown to be achieved by the well-known "string-pulling" strategy, graphically referring to pulling a string across the tunnel. Last, Chen *et al.* proposed an online peer offloading framework based on Lyapunov optimization and game theoretic approaches in [127], which enables small BSs cooperation to handle the spatially uneven computation workloads in the network.

4) Summary and Insight: The comparison of resource management schemes for multiuser MEC systems is provided in Table V. We draw several conclusions on resource allocation, MEC server scheduling and mobile cooperative computing as follows.

- Consider multiuser MEC systems with finite radio-and-computational resources. For system-level objectives, e.g., to minimize the sum mobile energy-consumption, the users with large channel gains and low local-computation energy consumption have higher priorities for offloading computation since they can contribute to larger energy savings. Too many offloading users, however, will cause severe inter-user interference of communication

and computation, which will, in turn, reduce the system revenue.

- To effectively reduce the sum computation latency of multiple users, the scheduling design for an MEC server should assign higher priorities to the users with more stringent latency requirements and heavy computation loads. Moreover, parallel computing can further boost the computation speed at the server.
- Scavenging the enormous amount of distributed computational resources can not only alleviate the network congestion, but also improves resource utilization and enables ubiquitous computing. This vision can be materialized by peer-to-peer mobile cooperative edge computing. The key advantages include short-range transmission via D2D techniques and computational resource and result sharing.

C. MEC Systems With Heterogeneous Servers

To enable ubiquitous edge computing, *heterogeneous MEC* (Het-MEC) systems were proposed in [128] comprising one central cloud and multiple edge servers. The coordination and interaction of multi-level central/edge clouds introduce many new research challenges and recently have attracted extensive relevant investigations on server selection, cooperation and computation migration, as discussed in the sequel.

1) Server Selection: For users served by a Het-MEC system, a key design issue is to determine the destination of computation offloading, i.e., either the edge or central cloud server. In [129], the server selection problem was studied for a multiuser system comprising a single edge server and a single central cloud. To maximize the total successful offloading probability, a heuristic scheduling algorithm was proposed to leverage both the low communication latency due to the proximity of the MEC server and the low computation latency arising from abundant computational resources at the central-cloud server. Specifically, when the computation load of the MEC server exceeds a given threshold, latency-tolerant tasks are offloaded to the central cloud to spare enough computational resources at the edge server for processing latency-sensitive tasks. In addition, [130] explored the problem of server selection over multiple MEC servers. The major challenge arises from the correlation between the amounts of the offloaded computation and selected edge servers for multiple users. To cope with this issue, a congestion game was formulated and solved to minimize the sum energy consumption of mobile users and edge servers. Most recently, a computation offloading framework that allows a mobile device to offload tasks to multiple MEC servers was proposed in [131], and semidefinite relaxation-based algorithms were proposed to determine the task allocation decisions and CPU frequency scaling.

2) Server Cooperation: Resource sharing via server cooperation can not only improve the resource utilization and increase the revenue of computing service providers, but also provide more resources for mobile users to enhance their user experience. This framework was originally proposed in [132], which includes components such as resource allocation, revenue management and service provider cooperation. First, resource allocation was optimized for cases with deterministic and random user information to maximize the total revenues. Second, considering self-interested cloud service providers, a distributed algorithm based on game theory was proposed to maximize service providers' own profits, which was shown to achieve the Nash equilibrium. This study was further extended in [133], which considered both the local and remote resource sharing. The former refers to resource sharing among different service providers within the same data center, while the latter one means the cooperation across different data centers. To realize the resource sharing and cooperation among different servers, a coalition game was formulated and solved by a game-theoretic algorithm with stability and convergence guarantees. Moreover, the recent work [134] proposed a new server cooperation scheme where edge servers exploit both the computational and storage resources by proactively caching computation results to minimize the computation latency. The corresponding task distribution problem was formulated as a matching game and solved by an efficient algorithm based on a proposed deferred-acceptance algorithm.

3) Computation Migration: In [135]–[137], apart from optimizing the offloading decisions, the authors also investigated the computation migration among different remote servers. Specifically, the computation migration over MEC servers was motivated by the mobility of offloading users. When a user moves closer to a new MEC server, the network controller can choose to migrate the computation to this server, or compute the task in the original server and then forward the results back to the user via the new server. The computation migration problem was formulated as an MDP problem based on a random-walk mobility model in [135]. It was shown that the optimal policy has a threshold-based structure, i.e., the migration should be selected only when the distance of two servers is bounded by two given thresholds. This work was further extended in [136] where the workload scheduling in edge servers was integrated with the service migration to minimize the average overall transmission and reconfiguration costs using Lyapunov optimization techniques. Another computation migration framework was proposed in [137], where the MEC server can either process offloaded computation tasks locally or migrate them to the central cloud server. An optimization problem was formulated to minimize the sum mobile-energy consumption and computation latency. This problem was solved by a heuristic two-stage algorithm, which first determines the offloading decision for each user by the semidefinite relaxation and randomization techniques, and then performs the resource allocation optimization for all the users.

4) Summary and Insight: Table VI provides the summary of resource management schemes for MEC systems with heterogeneous servers. The literature provides a set of insights on server selection, cooperation, and computation migration, described as follows.

- Consider MEC systems with multiple computation tasks and heterogeneous servers. To reduce the sum computation latency, it is desirable to offload latency-insensitive but computation-intensive tasks to remote central cloud server and latency-sensitive ones to the edge servers.

TABLE VI
THE COMPARISON OF PAPERS FOCUSING ON MEC SYSTEMS WITH HETEROGENEOUS SERVERS

Theme	Design Type	Design Objective	Reference	Proposed Solution
Server selection	Edge/central server selection	Successful offloading probability	[129]	Propose a heuristic server selection algorithm according to the deadline requirements
	Edge server selection	Energy	[130]	Formulate a congestion game and propose a distributed algorithm admitting the Nash equilibrium
	Multiple edge servers	Energy and latency	[131]	Propose semidefinite relaxation-based algorithms for task allocation decisions and computational frequency scaling
Server cooperation	Edge server cooperation	Revenue	[132]	Propose a distributed resource allocation algorithm admitting the Nash equilibrium
	Edge/remote server cooperation	Utility	[133]	Formulate a coalition game and propose a game-theoretic algorithm
	Edge server proactive caching	Latency	[134]	Study the distribution and proactive caching of computing tasks in MEC
Computation migration	Edge server migration	Cost	[135]	Propose a threshold-based computation migration scheme according to the distance
			[136]	Propose online workload scheduling and migration algorithms using Lyapunov optimization techniques
	Remote server migration	Energy and latency	[137]	Propose a heuristic two-stage algorithm including migration decision and resource allocation

- Server cooperation can significantly improve the computation efficiency and resource utilization at MEC servers. More importantly, it can balance the computation load distribution over the networks so as to reduce sum computation latency while the resources are better utilized. Moreover, the server cooperation design should consider temporal-and-spatial computation task arrivals and server's computation capacities, time-varying channels, and servers' individual revenue.
- Computation migration is an effective approach for mobility management in MEC. The decision of migrate-or-not depends on the migration overhead, distances between users and servers, channel conditions, and servers' computation capacities. Specifically, when a user moves far away from its original MEC server, it is preferred to migrate the computation to nearby servers.

D. Challenges

In the preceding subsections, we have conducted a comprehensive survey on the state-of-the-art resource management techniques for MEC systems. However, the progress is still in the infant stage and many critical factors have been overlooked for simplicity, which need to be addressed in future research efforts. In the following, we identify three critical research challenges for resource management in MEC that remain to be solved.

1) Two-Timescale Resource Management: In most existing works, e.g., [85], [86], [94], [117], [119], and [138], wireless channels were assumed to remain static during the whole task execution process for simplicity. Nevertheless, this assumption may be unreasonable when the channel coherence time is much shorter than the latency requirement. For instance, at a carrier frequency of 2GHz, the channel coherence time can be as small as 2.5ms when the speed is 100km/h. For some mobile applications such as the MMORPG game PlaneShift,[4] the acceptable response time is 440ms and the excellent latency is 120ms [139]. In such scenarios, the task offloading process may be across multiple channel blocks, necessitating the two-timescale resource management for MEC. This problem is very challenging even for a single-user MEC system with deterministic task arrivals [79].

2) Online Task Partitioning: For ease of optimization, existing literature tackling the task partitioning problems ignores the fluctuation of the wireless channels, and obtains the task partitioning decision before the start of the execution process. With such an offline task partitioning decision, the change of the channel condition may lead to inefficient or even infeasible offloading, which shall severely degrade the computation performance. To develop online task partitioning policies, one should incorporate the channel statistics into the formulated task partitioning problem, which may easily belong to an NP-hard problem even under a static channel. In [97] and [140], approximate online task partitioning algorithms were derived for applications with serial and tree-topology task-call graphs, respective, while solutions for general task models remain unexploited.

3) Large-Scale Optimization: The collaboration of multiple MEC servers allows their resources to be jointly managed for serving a large number of mobile devices simultaneously. However, the increase of the network size renders the resource management a large-scale optimization problem with respect to a large number of offloading decisions as well as radio-and-computational resource allocation variables. Conventional centralized joint radio-and-computational resource management algorithms require a huge amount of information and computation when applied to large-scale MEC systems, which will inevitably incur a significant execution delay and may whittle away the potential performance improvement, e.g., latency reduction, brought by the MEC paradigm. To achieve efficient resource management, it is required to design distributed low-complexity large-scale optimization algorithms with light signaling and computation overhead. Although the recent advancements in large-scale convex optimization [141] provide powerful tools for radio resource management, they cannot be directly applied to optimize the computation offloading decision due to its combinatorial and non-convex nature, which calls for new algorithmic techniques.

[4] http://www.planeshift.it/

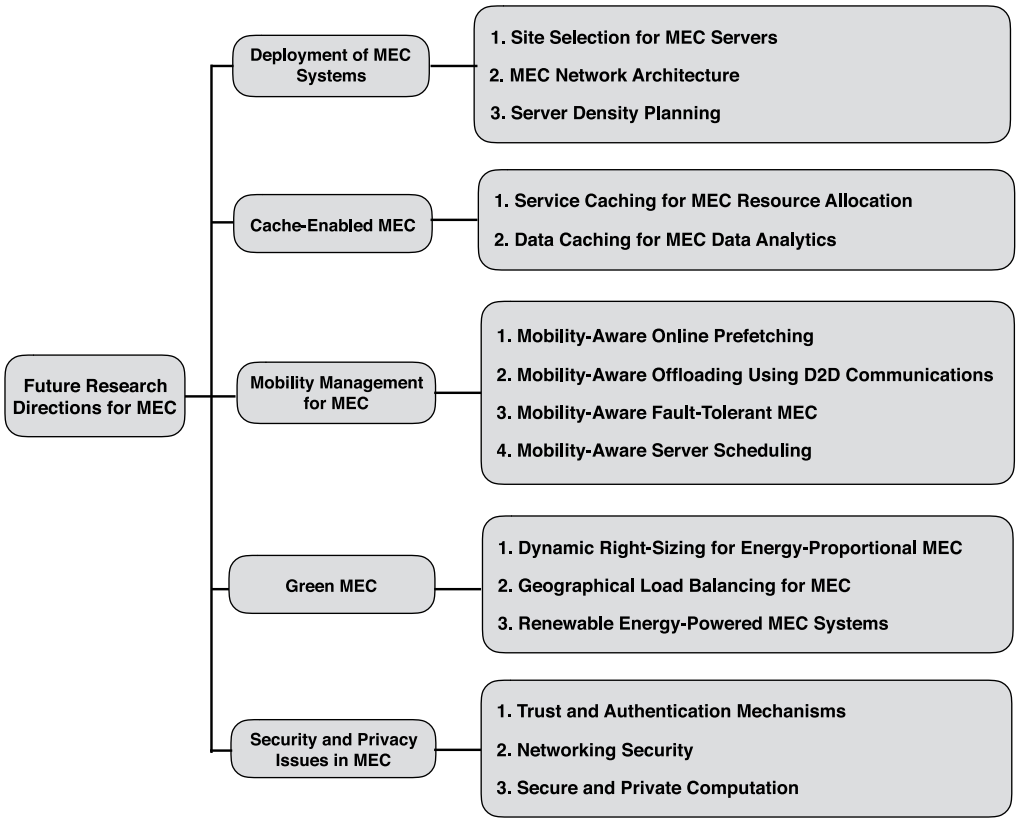


Fig. 7. Future research directions for MEC.

IV. Issues, Challenges, and Future Research Directions

Recent years have witnessed substantial research efforts on resource management for MEC as surveyed in the preceding section. However, there are lots of emerging research directions of MEC that are still largely uncharted. In this section, technical issues, challenges and research opportunities will be identified and discussed as summarized in Fig. 7, including the large-scale MEC system deployment, cache-enabled MEC, mobility management, green MEC and security-and-privacy issues in MEC.

A. Deployment of MEC Systems

The primary motivation of MEC is to shift the Cloud Computing capability to the network edges in order to reduce the latency caused by congestion and propagation delays in the core network. However, there is no formal definition of what an MEC server should be, and the server locations in the system are not specified. These invoke the site selection problems for MEC servers, which are significantly different from the conventional BS site selection problems, as the optimal placement of edge servers is coupled with the computational resource provisioning, and both of them are constrained by the deployment budget. Besides, the efficiency of an MEC system relies heavily on its architecture, which should account for various aspects such as workload intensity and communication rate statistics. In addition, it is critical for MEC vendors to determine the required server density for catering the service demand, which is closely related to the infrastructure deployment costs and marketing strategies. Nonetheless, the large-scale nature of MEC systems makes traditional simulation-based methods inapplicable, and thus solutions based on network-scale analysis are preferred. In this subsection, we will discuss three research problems related to MEC deployment, including the site selection for MEC servers, the MEC network architecture, and server density planning.

1) Site Selection for MEC Servers: Selecting the sites for MEC infrastructures, especially MEC servers, is the first step towards building up the MEC system. To make the cost-effective server-site selection, the system planners and administrators should account for two important factors: site rentals and computation demands. In general, given the system deployment budget, more MEC servers should be installed at regions with higher computation demands, such as business districts, commercial areas and densely populated areas. This, however, contradicts the cost requirement as such areas are likely to have high site rentals. Fortunately, thanks to the well-deployed telecom networks, it is a promising idea to install the MEC servers co-located with the existing infrastructures such as macro BSs, which is even more attractive for the telecom operators who would like to participate in the MEC market.

However, this would not solve all the problems. On one hand, due to the ever-increasing computation-quality requirement and ubiquitous smart devices, satisfactory user experience cannot be guaranteed due to the poor signal quality and congestion in the macro cells. For some applications,

e.g., smart home [142], it is desirable to move the computation capability even closer to the end users. This can be achieved by injecting some computational resources at small-cell BSs [70], [71], which are low-cost and small-size BSs. Despite the potential benefits, there are still obstacles on the way:

- First, due to physical limitations, the computation capabilities of such kind of MEC servers will be much smaller than those at macro BSs, making it challenging to handle computation-intensive tasks. One feasible solution is to build a hierarchical network architecture for MEC systems comprising MEC servers with heterogeneous communication-and-computation capabilities as detailed in the sequel.
- Second, some of the small-cell BSs may be self-deployed by the home users, and many femto BS owners may not have the motivation to collaborate with MEC vendors. To overcome this issue, MEC vendors need to design a proper incentive mechanism in order to stimulate the owners of small-cell BSs for renting the sites.
- Moreover, deploying MEC servers at small-cell BSs may incur security problems as they are easy-to-reach and vulnerable to external attacks, which shall degrade the levels of reliability.

On the other hand, the computation hot spots do not always coincide with the communication hot spots. In other words, for some of the computation hot spots, there exists no available communication infrastructure (either macro or small-cell BS). For these circumstances, we need to deploy edge servers with wireless transceivers by properly choosing new locations.

Besides, the site selection for MEC servers is dependent on the computational resource-allocation strategy, which poses extra challenges compared to the conventional BS site selection. Intuitively, concentrating the computational resources at a few MEC servers can help save the site rentals. However, this comes at the prices of potential degradation of the service coverage and communication quality. In addition, the optimal computational resource allocation should take into account both site rentals and computation demands. For example, for an MEC server at a site with a high site rental, it is preferred to allocate huge computational resource and thus serve a large number of users, for achieving the high revenue. Hence, a joint site selection and computational resource provisioning problem needs to be solved before deploying MEC systems.

2) MEC Network Architecture: The promotion of MEC does not mean the extinction of the *data-center networks* (DCNs). Instead, future mobile computing networks are envisioned to be consisted of three layers as shown in Fig. 8, i.e., cloud, edge (a.k.a. fog layer), and the service subscriber layer [128], [143]. While the cloud layer is mature and well-deployed, there is still some flexibility and uncertainty in designing the edge layer.

By analogy to the *heterogeneous networks* (HetNets) in cellular systems, it is intuitive to design the Het-MEC systems, which consist of multiple tiers. Specifically, the MEC servers in different tiers have distinct computation and communication capabilities. Such kinds of hierarchical MEC system structures can not only preserve the advantage of efficient transmission offered by HetNets, but also possess strong ability to handle the peak computation workloads by distributing them across different tiers [144]. However, the computation capacity provisioning problem is highly challenging and remains unsolved, as it should account for many different factors, such as the workload intensity, communication cost between different tiers, workload distribution strategies, etc.

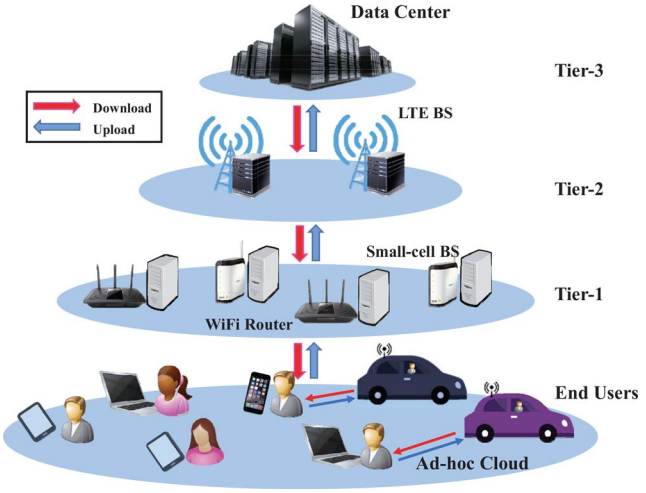


Fig. 8. A 3-tier heterogeneous MEC system. Tier-1 servers are located in close proximity to the end users, such as at WiFi routers and small-cell BSs, which are of relatively small computation capabilities. Tier-2 servers are deployed at LTE BSs with moderate computation capabilities. Tier-3 servers are the existing Cloud Computing infrastructures, such as data centers.

Another thrust of research efforts focuses on exploiting the potential of the service subscriber layer, and utilizing the undedicated computational resources, e.g., laptops, smart phones, and vehicles, overlaid with dedicated edge nodes. This paradigm is termed as the *Ad-hoc mobile cloud* in literature [145]–[148]. The ad-hoc mobile cloud enjoys the benefits of amortizing the stress of MEC systems, increasing the utilization of the computational resources, and reducing the deployment cost. However, it also brings difficulties in resource management and security issues due to its ad-hoc and self-organized nature.

3) Server Density Planning: As mentioned in Section IV-A2, the MEC infrastructure may be a combination of different types of edge servers, which provides various levels of computation experience and contributes different deployment costs. Hence, it is critical to determine the number of edge nodes as well as the optimal combination of different types of MEC servers with a given deployment budget and computation demand statistics. Conventionally, this problem can only be addressed by numerical simulations, which is time-consuming and has poor scalability. Fortunately, owing to the recent development of *stochastic geometry theory* and its successful applications in performance analysis for wireless networks [149]–[152], as well as the similarity between Het-MEC systems and HetNets, it is feasible to conduct performance analysis for MEC systems using techniques from stochastic geometry theory. Such analysis of MEC systems should address the following challenges: 1) The timescales of computation and wireless channel coherence time may be different [79], [102], which makes existing results for wireless networks not readily

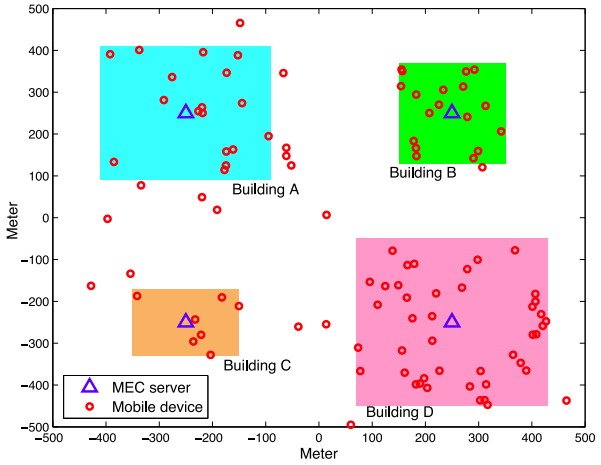


Fig. 9. Illustration of the clustering behavior of the computation demands. The mobile devices requesting for MEC services will be more concentrated around the MEC servers.

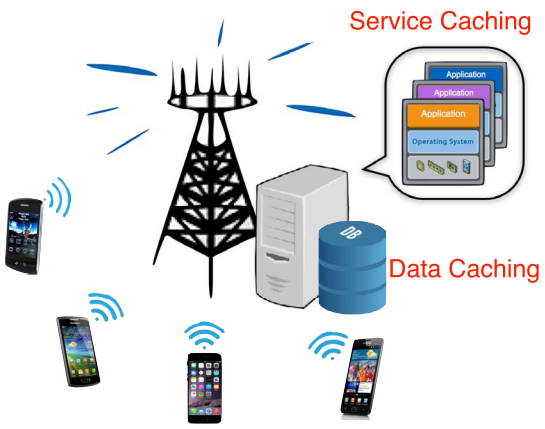


Fig. 10. Cache-enabled MEC systems.

applicable for MEC systems. One possible solution is to combine the Markov chain and stochastic geometry theories to capture the steady behavior of computations. 2) The computation offloading policy will affect the radio resource management policy, which should be taken into consideration. 3) The computation demands are normally non-uniformly distributed and clustered (see Fig. 9), prohibiting the use of the *homogeneous Poisson point process* (HPPP) model for edge servers and service subscribers. It thus calls for the investigation of more advanced point processes, e.g., the Ginibre *α-determinantal point process* (DPP), to capture the clustering behaviors of edge nodes [153].

B. Cache-Enabled MEC

It has been predicted by Cisco that mobile video streaming will occupy up to 72% of the entire mobile data traffic by 2019 [154]. One unique property of such services is that the content requests are highly concentrated and some popular contents will be asynchronously and repeatedly requested. Motivated by this fact, *wireless content caching* or *FemtoCaching* was proposed in [155]–[158] to avoid frequent replication for the same contents by caching them at BSs. This technology has attracted extensive attention from both academia and industry due to its striking advantages on reducing content acquisition latency, as well as relieving heavy overhead burden of the network backhaul. While caching is to move popular contents close to end users, MEC is to deploy edge servers to handle computation-intensive tasks for edge users to enhance user experience. Note that these two techniques seem to target for diverse research directions, i.e., one for popular content delivery and the other for individual computation offloading. However, they will be integrated seamlessly in this subsection and envisioned to create a new research area, namely, the *cache-enabled MEC*.

Consider the novel cache-enabled MEC system shown in Fig. 10. In such systems, the MEC server can cache several application services and their related database, called *service caching* (or service placement [159]) and *data caching*, respectively, and handle the offloaded computation from multiple users. To efficiently reduce the computation latency, several key and interesting problems need to be solved, which are described in the following with potential solutions.

1) Service Caching for MEC Resource Allocation: Unlike the central cloud server that is always assumed with huge and diverse resources (e.g., computing, memory and storage), the current edge server has much less resources, making it unable to accommodate all users' computation requests. On the other hand, different mobile services require different resources, based on which, they can be classified into CPU-hungry (e.g., cloud chess and VR), memory-hungry (e.g., online MATLAB), and storage-hungry (e.g., VR) applications. Such a mismatch between resource and demand introduces a key challenge on how to allocate heterogeneous resources for service caching.

Note that similar problems have been investigated in conventional Cloud Computing systems [160]–[163], termed as *VM placement*, as well as MCC systems [159]. Specifically, Tordsson *et al.* [160] proposed a novel architecture for VM management and optimized the VM placement over multiple clouds to reduce the deployment costs and improve user experience, given constraints on hardware configuration, the number of VMs as well as load balancing. Similar VM-placement problems were also investigated in [161] and [162] for maximizing the energy savings of cloud servers and in [163] for different cloud scheduling strategies. Recently, Yang *et al.* [159] extended the VM placement idea to MCC systems and studied the joint optimization of service caching/placement over multiple clouds and load dispatching for end users' requests. As a result, one efficient algorithm was proposed to minimize both the computation latency and service placement transition cost. These works, however, cannot be directly applied to design efficient service caching policies for MEC systems, since it should take into account more refined information including users' location, preference, experience as well as edge servers' capacities in terms of the memory, storage and VM instance. To this end, two possible approaches are described as follows.

The first one is *spatial popularity-driven service caching*, referring to caching different combinations and amounts of services in different MEC servers according to their specific

locations and surrounding users' common interests. This idea is motivated by the fact that users in one small region are likely to request similar computing services. For example, visitors in a museum tend to use AR for better sensational experience. Thus, it is desirable to cache multiple AR services at the MEC server of this region for providing the real-time service. To achieve the optimal spatial service caching, it is essential to construct a *spatial-application popularity distribution model* for characterizing the popularity of each application over different locations. Based on this, we can design resource-allocation policies using various optimization algorithms, e.g., the game theory and convex optimization techniques.

An alternative approach is *temporal popularity-driven service caching*. The main idea is similar to that of the spatial counterpart, but it exploits the popularity information in the temporal domain, since the computation requests also depend on the time period. One example is that users are apt to play mobile cloud gaming after dinner. This kind of information will suggest MEC operators to cache several gaming services during this typical period for handling the huge computation loads. One disadvantage of this temporal-based approach is the additional server cost resulted from frequent *cache-and-tear* operations since popularity information is time-varying and MEC servers possess finite resources.

2) Data Caching for MEC Data Analytics: Many modern mobile applications involve intensive computation based on data analytics, e.g., ranking and classification. Take VR as an instance. It creates an imaginary environment similar to the real world by generating realistic images, sounds and other sensations for enhancing users' experience. Achieving this end is nontrivial as it requires the MEC server to finish multiple complicated processes within the ultra-short duration (e.g., 1ms), such as recognizing users' actions via pattern recognition, "understanding" users' requests via data mining, as well as rendering virtual settings via video streaming or other sensation techniques [164]. All the above data-analytics based techniques should be supported by comprehensive database, which, however, imposes extremely heavy burden on the edge server storage. This challenge can be relieved by intelligent data caching that only reserves frequently-used database. From another perspective, caching parts of computation-result data that is likely to be reused by others can further boost the computation performance of the entire MEC system. One typical example is mobile cloud gaming, which enables fast and energy-efficient gaming by shifting game computing engines from mobiles to edge servers and supporting real-time gaming by game video streaming. Thus, it emerges as a leading technique for next generation mobile computing infrastructures [139]. Since certain game rendered videos, e.g., gaming scenes, can be reused by other players, caching these computation results would not only significantly reduce the computation latency of the players with the same computation request, but also ease the computation burden for edge servers. Similar idea has been proposed in [165], which investigated collaborative multi-bitrate video caching and processing in MEC.

For MEC data caching at a single edge server, one key problem is *how to balance the tradeoff between massive database and finite storage capacity*. Unlike FemtoCaching networks where content (data) caching mainly introduces a new multiple-access mechanism termed as cache-enabled access [166], data caching in MEC systems brings about manifold effects on the computation accuracy, latency and edge server-energy consumption, which, however, have not been characterized in existing literature. This calls for model building research efforts for accurately quantifying the mentioned effects for various MEC applications. Furthermore, it is also essential to establish a practical *database popularity distribution model* that is able to statistically characterize the usage of each database set for different MEC applications. Based on the above models, the said tradeoff can be achieved by solving an optimization problem that maximizes the achievable QoS and minimizes the storage cost in MEC systems simultaneously.

The above framework can be further extended to MEC systems with multiple servers where each server can serve multiple users and each user can offload computation to multiple edge servers. The fundamental problem is similar to that of the cache-enabled HetNets [167], that is, how to *spatially* distribute the database over heterogeneous edge servers under both storage and computation-load constraints on each of them, for increasing network-wide revenue. Intuitively, for each MEC server, it is desirable to spare more storage to cache the database of the most popular applications in its cell, and it also needs to utilize partial storage to accommodate less popular ones, whose computation performance will be further improved by cooperative caching in different MEC servers. Moreover, the performance of large-scale cache-enabled MEC networks can be analyzed using stochastic geometry by modeling nearby users as clusters [168].

C. Mobility Management for MEC

Mobility is an intrinsic trait of many MEC applications, such as AR assisted museum tour to enhance experience of visitors. In these applications, the movement and trajectory of users provide location and personal preference information for the edge servers to improve the efficiency of handling users' computation requests. On the other hand, mobility also poses significant challenges for realizing ubiquitous and reliable computing (i.e., without interruptions and errors) due to the following reasons. First, MEC will be typically implemented in the HetNet architecture comprising of multiple macro, small-cell BSs and WiFi APs. Thus, users' movement will call for frequent handovers among the small-coverage edge servers as shown in Fig. 11, which is highly complicated due to the diverse system configurations and user-server association policies. Next, users moving among different cells will incur severe interference and pilot contamination, which shall greatly degrade the communication performance. Last, frequent handovers will increase the computation latency and thus deteriorate users' experience.

Mobility management has been extensively studied for traditional heterogeneous cellular networks [169]–[171]. In these prior works, users' mobility is modeled by the connectivity probability or the link reliability according to such information as the users' moving speeds. Based on such models,

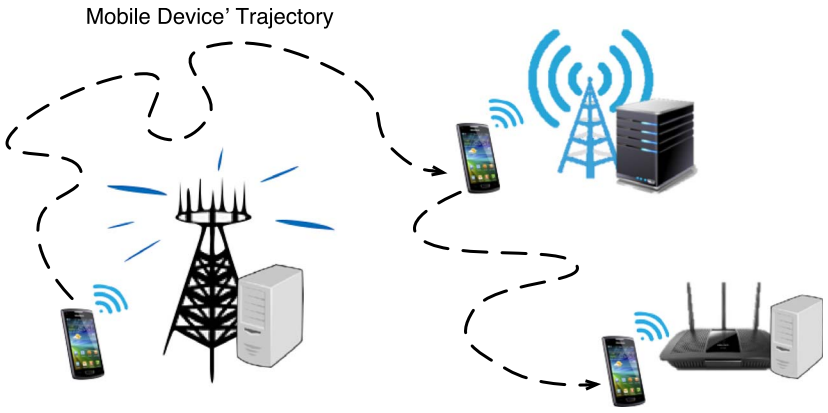


Fig. 11. Mobility management for MEC.

dynamic mobility management has been proposed to achieve high data rate and low bit-error rate. However, these policies cannot be directly applied for MEC systems with moving users, since they neglect the effects of the computational resources at edge servers on the handover policies. Recent works in [172]–[175] have made initial efforts to design mobility-aware MEC systems. Specifically, the inter-contact time and contact rate were defined in [172] to model users' mobility. An opportunistic offloading policy was then designed by solving a convex optimization problem for maximizing the successful task offloading probability. Alternatively, to account for the mobility, the number of edge servers that users can access was modeled by an HPPP in [173]. Then, the offloading decision was optimized by addressing the formulated MDP problem to minimize the offloading cost including mobile-energy consumption, latency and failure penalty. Other mobility models were also proposed in [174] and [175], which characterize the mobility by a sequence of networks that users can connect to and a two-dimensional location-time workflow, respectively. In addition, mobility management for MEC was integrated with traffic control in [176] to provide better experience for users with latency-tolerant tasks via designing intelligent cell association mechanisms. In [158], edge caching was integrated with mobility prediction in Follow-Me Cloud for enhancing the content-caches migration located at the edges. Recent proposals on mobility-aware wireless caching in [177] also provided valuable guidelines on mobility management in MEC systems.

Note that most of the existing works focused on optimizing mobility-aware server selection. However, to achieve better user experience and higher network-wide profit, the offloading techniques at mobile devices and scheduling policies at MEC servers should be jointly considered. This introduces a set of interesting research opportunities with some described as follows.

1) Mobility-Aware Online Prefetching: In practice, the full information of the user trajectory may be unavailable. Conventional design for mobile computation offloading will fetch a computation task to another server only when it is handoverred. This mechanism requires excessive fetching of a large volume of data for handover and thus brings long fetching latency. Moreover, it also causes heavy loads on the MEC network. One promising solution to handle this issue is to leverage the statistical information of the user trajectory and prefetch parts of future computation data to potential servers during the server-computation time, referred to as *online prefetching* [178]. This technique can not only significantly reduce the handover latency via mobility prediction, but also enable energy-efficient computation offloading by enlarging the transmission time. However, it also encounters several challenges with two most critical ones described as follows. The first challenge arises from the trajectory prediction. Accurate prediction can allow seamless handovers among edge servers and reduce the prefetching redundancy. Achieving it, however, requires precise modeling and high-complexity ML techniques, e.g., Bayesian, reinforcement and deep learning. For example, the trajectory of a typical visitor in a museum can be predicted according to his own interest-information and statistical route-information of some previous visitors with similar interests that can be obtained by ML algorithms. Therefore, it is important to balance the tradeoff between the modeling accuracy and computation complexity. The second challenge lies in the selection of the prefetched computation data. To maximize the successful offloading probability of edge users, the computation-intensive components should be prefetched earlier with adaptive transmission power control in dynamic fading channels.

2) Mobility-Aware Offloading Using D2D Communications: D2D communications was first proposed in [179] to improve the network capacity and alleviate the data traffic burden in cellular systems. This paradigm can also be used to handle the user mobility problems in MEC systems [121], which creates numerous D2D communication links. These links allow the computation of a user to be offloaded to its nearby users which have more powerful computation capabilities. The short-range communication offered by D2D links reduces energy consumption of data transmission as well. However, user mobility brings new design issues as follows. The first one is how to exploit the advantages of both D2D and cellular communications. One possible approach is to offload the computation-intensive data to the edge servers at BSs that have huge computation capabilities in order to reduce the server-computing time; while the components of large data sizes and strict computation requirements should be

fetched to nearby users via D2D communications for higher energy efficiency. Next, the selection of surrounding users for offloading should be optimized to account for users' mobility information, dynamic channels and heterogeneous users' computation capabilities. Last, massive D2D links will introduce severe interference for reliable communications. This issue is more complicated in the mobility-based MEC systems due to the fast-changing wireless fading environments. Hence, advanced interference cancellation and cognitive radio techniques can be applied for MEC systems, together with mobility prediction to increase the offloading rate and reduce the service latency.

3) Mobility-Aware Fault-Tolerant MEC: User mobility poses significant challenges for providing reliable MEC services due to dynamic environments. Computation offloading may fail due to intermittent connections and rapid-changing wireless channels. The induced failure is catastrophic for the latency-sensitive and resource-demanding applications. For instance, AR-based museum video guide aims to provide fluent and fancy virtual sensations for visitors, and the disruption or failure of video streaming due to intermittent connections would upset visitors. Another example is the military operation which always requires fast and ultra-reliable computation, even in high-mobility environments. Any computation failure would bring serious consequences. These facts necessitate the design for mobility-aware fault-tolerant MEC systems [180]–[182], with three major and interesting problems illustrated as follows, including fault prevention, fault detection and fault recovery. Fault prevention is to avoid or prevent MEC fault by backing up extra stable offloading links. Macro BSs or central clouds can be chosen as protection-clouds, since they have large network coverage that allows continuous MEC service. The key design challenges lie in how to balance the tradeoff between QoS (i.e., the failure probability) and energy consumption due to extra offloading links for the single-user case, and how to allocate protection-clouds for multiuser MEC applications. Next, fault detection is to collect fault information, which can be realized by setting intelligent timing checks or receiving feedbacks for MEC services. In addition, channel and mobility estimation techniques can also be applied to estimate the fault so as to reduce the detection time. Last, for detected MEC faults, recovery approaches should be performed to continue and accelerate the MEC service. The suspended service can be switched to more reliable backup wireless links with adaptive power control for higher-speed offloading. Alternative recovery approaches include migrating the workloads to neighboring MEC systems directly or through ad-hoc relay nodes as proposed in [182].

4) Mobility-Aware Server Scheduling: For multiuser MEC systems, traditional MEC server scheduling serves users according to the offloading priority order that depends on users' distinct local computing information, channel gains and latency requirements [82]. However, this static scheduling design cannot be directly applied for the multiuser MEC systems with mobility due to dynamic environments, e.g., time-varying channels and intermittent connectivities. Such dynamics motivate the design of adaptive server scheduling that regenerates the scheduling order from time to time, incorporating the real-time user information. In such adaptive scheduling mechanisms, users with worse conditions will be allocated with higher offloading priorities to meet their computing deadlines. Another potential approach is to design mobility-aware offloading priority function by the following two steps. The first step is to accurately predict users' mobility profiles and channels, where the major challenge is how to reflect the mobility effects and re-define the offloading priority function. The second step is resource reservation that can enhance the server scheduling performance [183], [184]. Specifically, to guarantee the QoS of latency-sensitive and high-mobility users, MEC servers can reserve some dedicated computational resources and provide reliable computing service for such users. While for other latency-tolerant users, the MEC server can perform on-demand provisioning. For such a hybrid MEC server provisioning scheme, the server scheduling can be optimized for serving the maximum number of users with QoS guarantees, as well as maximizing MEC servers' revenue.

D. Green MEC

MEC servers are small-scale data centers, each of which consumes substantially less energy than the conventional cloud data center. However, their dense deployment pattern raises a big concern on the system-wide energy consumption. Therefore, it is unquestionably important to develop innovative techniques for achieving green MEC [185], [186]. Unfortunately, designing green MEC is much more challenging compared to green communication systems or green DCNs. Compared to green communication systems, the computational resource needs to be managed to guarantee satisfactory computation performance, making the traditional green radio techniques not readily applicable. On the other hand, the previous research efforts on green DCNs have not considered the radio resource management, which makes them not suitable for green MEC. Besides, the highly unpredictable computation workload pattern in MEC servers poses another big challenge for resource management in MEC systems, calling for advanced estimation and optimization techniques. In this subsection, we will introduce different approaches on designing green MEC systems, including dynamic right-sizing for energy-proportional MEC, *geographical load balancing* (GLB) for MEC, and MEC systems powered by renewable energy.

1) Dynamic Right-Sizing for Energy-Proportional MEC: The energy consumption of an MEC server highly depends on the utilization radio [see Eq. (5)]. Even when the server is idle, it still consumes around 70% of the energy as it operates at the full speed. This fact motivates the design of *energy-proportional* (or *power-proportional*) servers, i.e., the energy consumption of a server should be proportional to its computation load [187]. One way to realize energy-proportional servers is to switch off/slow down the processing speeds of some edge servers with light computation loads. Such an operation is termed as *dynamic right-sizing* in the literature on green DCNs [188]. However, along with the potential energy savings, toggling servers between the active and sleep modes

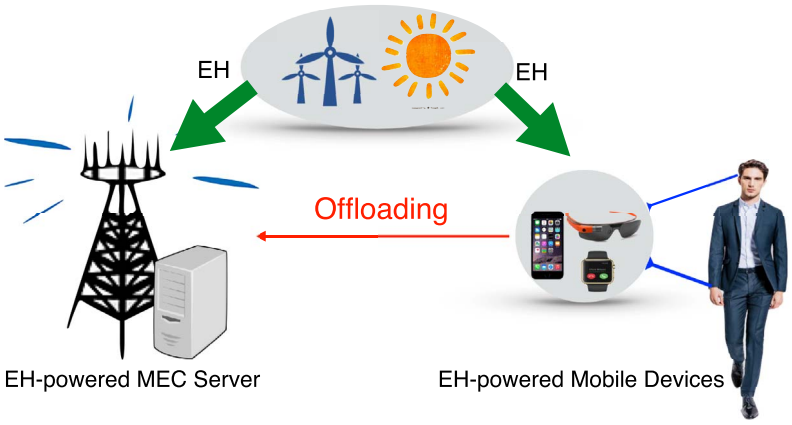


Fig. 12. Renewable energy-powered MEC systems.

could bring detrimental effects. First of all, it will incur the switching energy cost and application data-migration latency. Also, user experience may be degraded due to the less amount of allocated computational resources, which may, in turn, reduce the operator's revenue. Besides, the risk associated with server toggling as well as the *wear-and-tear* cost of the servers might be increased, which can in turn increase the maintenance costs of MEC vendors. As a result, switching off the edge servers in a myopic manner is not always beneficial.

In order to make an effective decision on dynamic right-sizing, the profile of computation workload at each edge server should be accurately forecasted. In conventional DCNs, this can be achieved rather easily as the workload at each data center is an aggregation of the computation requests across a large physical region, e.g., several states in the United States, which is relatively stable so that it can be estimated by referring to the readily available historical data at the data centers. However, for MEC systems, the serving area of each edge server is much smaller, and hence its workload pattern is affected by many factors, such as the location of the server, time, weather, the number of nearby edge servers, and user mobility. This leads to a fast-changing workload pattern, and requires more advanced prediction techniques. Moreover, online dynamic right-sizing algorithms that require less future information need to be developed.

2) Geographical Load Balancing for MEC: GLB is another key technique for green DCNs [189], [190], which leverages the *spatial diversities* of the workload patterns, temperatures, and electricity prices, to make workload routing decision among different data centers. This technique can also be applied to MEC systems. For instance, a cluster of MEC servers can coordinate together to serve a mobile user, i.e., the tasks can be routed from the edge server located in a hot spot (such as a restaurant) to a nearby edge server with light workload (such as the one in a park). On one hand, this helps to improve the energy efficiency of the lightly-loaded edge servers as well as user experience. On the other hand, it can prolong the battery lives of mobile devices, as offloading the tasks through the nearby server could save transmission energy. It is worthwhile to note that the implementation of GLB requires efficient resource management techniques at edge servers, such as dynamic right-sizing and VM management [191]–[194].

Meanwhile, there are many factors to be incorporated when applying GLB in MEC environments. Firstly, since the migrated tasks should go through the cellular core network, the network congestion state should be monitored and considered when making the GLB decisions. Secondly, to enable seamless task migration, a VM should be migrated/set up in another edge server beforehand, which may cause additional energy consumption. Thirdly, the mutual interests of MEC operators and edge computing service subscribers should be carefully considered when performing GLB, due to the tradeoff between the energy savings and latency reduction. Last but not least, the existence of conventional Cloud Computing infrastructures endows the edge servers with an extra option of offloading the latency-critical and computation-intensive tasks to remote cloud data centers, creating a new design dimension and further complicating the optimization.

3) Renewable Energy-Powered MEC Systems: Traditional grid energy is normally generated by coal-fired power plants. Hence, powering mobile systems with grid energy inevitably causes a huge amount of carbon emission, which opposes the target of green computing. Off-grid renewable energy, such as solar radiation and wind energy, recently, has emerged as a viable and promising power source for various IT systems thanks to the recent advancements of *energy harvesting* (EH) techniques [195], [196]. This fact motivates the design of innovative MEC systems, called renewable energy-powered MEC systems, which are shown in Fig. 12 comprising both EH-powered MEC servers and mobile devices. On one hand, as the MEC servers are expected to be densely-deployed and have low power consumption similar to that of small-cell BSs [197], it is reasonable and feasible to power the MEC infrastructures with the state-of-the-art EH techniques. On the other hand, the mobile devices can also get benefits from using renewable energy as EH is able to prolong their battery lives, which is one of the most favorable features for mobile phones [198]. Besides, the use of renewable energy sources eliminates the need of human intervention such as replacing/recharging the batteries, which is difficult if not impossible

for certain types of application scenarios where the devices are hard and dangerous to reach. Meanwhile, these advantages of using renewable energy are accompanied with new design challenges.

A fundamental problem to be addressed for renewable energy-powered MEC systems is the *green energy-aware resource allocation and computation offloading*. Instead of minimizing the energy consumption subject to satisfactory user experience, the design principle for the renewable energy-powered MEC systems should be changed to optimizing the achievable performance given the renewable energy constraint, as the renewable energy almost comes for free. Also, with renewable energy supplies, the *energy side information* (ESI), which indicates the amount of available renewable energy, will play a key role in the decision making. Initial investigations on renewable energy-powered MEC systems were conducted in [199] and [200], which focused on EH-powered MEC servers and EH-powered mobile devices, respectively. For EH-powered MEC servers, the system operator should decide the amount of workload required to be offloaded from the edge server to the central cloud, as well as the processing speed of the edge server, according to the information of the core network congestion state, computation workload, and ESI. This problem was solved by a learning-based online algorithm in [199]. While for EH-powered mobile devices, a dynamic computation offloading policy has been proposed in [200] using Lyapunov optimization techniques based on both the CSI and ESI. However, these two works only considered small-scale MEC systems that consist of either one edge server (in [199]) or one mobile device (in [200]). Thus, they cannot provide a comprehensive solution for large-scale MEC systems.

For large-scale MEC systems where multiple MEC servers are deployed across a large geographic region, the concept of GLB could be modified as the *green energy-aware GLB* to optimize the MEC systems by further utilizing the spatial diversity of the available renewable energy. This idea was originally proposed for green DCNs, where the *"follow the renewables"* routing scheme offers a huge opportunity in reducing the grid energy consumption [189], [201]–[204]. Moreover, as mentioned before, there exist significant differences between MEC systems and conventional DCNs in terms of the wireless channel fluctuation and resource-management design freedom of system operators. These factors make the offloading decision making for the green energy-aware GLB in MEC systems much more complicated, as it needs to consider the CSI and ESI in the whole system.

The randomness of renewable energy may introduce the offloading unreliability and risks of failure, bringing about a major concern for using renewable energy to power MEC systems. Fortunately, there are several potential solutions to circumvent this issue as described below.

- First, thanks to the low deployment cost, renewable energy-powered edge servers can be densely deployed over the system to provide more offloading opportunities for the users. The resultant overlapping serving areas offer the offloading diversity in the available energy to avoid performance degradation. A similar idea has been proposed for EH cooperative communication systems in [205].
- Second, the chance of energy shortage can be reduced by properly selecting the renewable energy sources. It was found in [189] that solar energy is more suitable for workloads with a high *peak-to-mean ratio* (PMR), while wind energy fits better for workloads with a small PMR. This provides guidelines for renewable energy provisioning for edge servers.
- Third, MEC servers can be powered by hybrid energy sources to improve reliability [206]–[208], i.e., powered by both the electric grid and the harvested energy. Also, equipping *uninterrupted power supply* (UPS) units at the edge servers can provide a short period of stable energy supply when green energy is in deficit, and it can be recharged when the surrounding energy condition returns to a good state.
- Moreover, *wireless power transfer* (WPT), which charges mobile devices using RF wave [209], [210], is a newly-emerged solution that enables wireless charging and extends the battery life. This technique has been provided in modern mobile phones such as Samsung Galaxy S6. In renewable energy-powered MEC systems, the edge servers can be powered by WPT when the renewable energy is insufficient for reliability [211]. This technology also applies to the computation offloading for mobile devices in MEC systems [81] and data offloading for collaborative mobile clouds [212]. However, novel energy beamforming techniques are needed to increase the charging efficiency. Moreover, due to the doubly near-far problem in wireless powered systems, it requires a delicate scheduling to guarantee fairness among multiple mobile devices.

E. Security and Privacy Issues in MEC

There are increasing demands for secure and privacy-preserving mobile services. While MEC enables new types of services, its unique features also bring new security and privacy issues. First of all, the innate heterogeneity of MEC systems makes the conventional trust and authentication mechanisms inapplicable. Second, the diversity of communication technologies that support MEC and the software nature of the networking management mechanisms bring new security threats. Besides, secure and private computation mechanisms become highly desirable as the edge servers may be an eavesdropper or an attacker. These motivate us to develop effective mechanisms as described in the following.

1) Trust and Authentication Mechanisms: Trust is an important security mechanism in almost every mobile system, behind which, the basic idea is *to know the identity of the entity that the system is interacting with*. Authentication management provides a possible solution to ensure "trust" [213]. However, the inherent heterogeneity of MEC systems, i.e., different types of edge servers may be deployed by multiple vendors and different kinds of mobile devices coexist, makes the conventional trust and authentication mechanisms designed for Cloud Computing systems inapplicable. For example, the

reputation-based trust model will lead to severe trust threats in MEC systems, as demonstrated in [214]. This fact calls for a unified trust and authentication mechanism that is able to assess the reliability of edge servers and identify the camouflaged edge servers. Besides, within the mobile network, there will be a large number of edge servers serving massive mobile devices. This makes the trust and authentication mechanism design much more complicated compared with that in conventional Cloud Computing systems, since edge servers are of small computation capabilities and designed to enable latency-sensitive applications. Therefore, it is critical to minimize the overhead of authentication mechanisms and design distributed policies [215], [216].

2) Networking Security: The communication technologies to support MEC systems, e.g., WiFi, LTE and 5G, have their own security protocols to protect the system from attacks and intrusions. However, these protocols inevitably create different trust domains. The first challenge of networking security in MEC systems comes from the difficulties in the distribution of credentials, which can be used to negotiate session keys among different trust domains [213]. In existing solutions, the certification authority can only distribute the credentials to all the elements located within its own trust domain [213], making it hard to guarantee the privacy and data integrity for communications among different trust domains. To address this problem, we can use the cryptographic attributes as credentials in order to exchange session keys [217], [218]. Also, the concept of federated content networks, which defines how multiple trust domains can negotiate and maintain inter-domain credentials [219], can be utilized.

Besides, techniques such as SDN and NFV are introduced to MEC systems to simplify the networking management as well as to provide isolation [5]. However, these techniques are softwares by nature and thus vulnerable [220], [221]. Moreover, the large number of devices and entities in MEC systems increase the chance of successfully attacking a single device, which provides means to launch an attack to the whole system [222]. Therefore, novel and robust security mechanisms, such as hypervisor introspection, run-time memory analysis, and centralized security management [223], are needed to guarantee a secure networking environment for MEC systems.

3) Secure and Private Computation: Migrating computation-intensive applications to the edge servers is the most important function and motivation of building MEC systems. In practice, the task input data commonly contains sensitive and private information such as personal clinical data and business financial records. Therefore, such data should be properly pre-processed before being offloaded to edge servers, especially the untrusted ones, in order to avoid information leakage. In addition to information leakage, the edge servers may return inaccurate and even incorrect computation results due to either software bugs or financial incentives, especially for tasks with huge computation demands [224]. To achieve secure and private computation, it is highly preferred that the edge platforms can execute the computation tasks without the need of knowing the original user data and the correctness of the computation results can be verified, which can be realized by encryption algorithms and verifiable computing techniques [225]. An interesting example of secure computation mechanisms for LP problems was developed in [224], where the LP problem is decomposed into the public-owned solvers and the private-owned data. By using a privacy-preserving transformation, the customer offloads the encrypted private data for cloud execution, and the server returns the results for the transformed LP problem. A set of necessary and sufficient conditions for verifying the correctness of the results were developed based on duality theory. Upon receiving the correct result, the clients can map back the desired solution for the original problem using the secret transformation. This method of result validation achieves a big improvement in computation efficiency via high-level LP computation compared to the generic circuit representation, and it incurs close-to-zero additional overhead on both the client and cloud server, which provides hints to develop secure and private computation mechanisms for other cloud applications.

V. Standardization Efforts and Use Scenarios of MEC

Standardization is an indispensable step for successful promotion of a new technology, which documents the consensus among multiple players and defines voluntary characteristics and rules in a specific industry. Due to the availability of structured methods and reliable data, standardization helps to promote innovation and disseminate groupbreaking ideas and knowledge about cutting-edge techniques. More importantly, standardization can build customer trust in products, services and systems, which helps to develop favorable market condition. The technical standards for MEC are being developed by ETSI, and a new *industry specification group* (ISG) was established within ETSI by Huawei, IBM, Nokia Networks, NTT DOCOMO and Vodafone. The aim of the ISG is to build up a standardized and open environment, which will allow the efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor MEC platforms [226]. In September 2014, an introductory technical white paper on MEC was published by ETSI, which defined the concept of MEC, proposed the referenced MEC platform, as well as pointed out a set of technical requirements and challenges for MEC [5]. Also, typical use scenarios and their relationships with MEC have been discussed. These aspects have also been documented in the ETSI specifications in 2015 [46], [227], [228], [242]. Most recently, ETSI has announced six *Proofs of Concepts* (PoCs) that were accepted by the MEC ISG in MEC World Congress 2016, which will assist the strategic planning and decision-making of organizations, as well as help to identify which MEC solutions may be viable in the network [229]. This provides the community with confidence in MEC and will accelerate the pace of the standardization. It is interesting to note that, in this congress, the ETSI MEC ISG has renamed *Mobile Edge Computing* as *Multi-access Edge Computing* in order to reflect the growing interest in MEC from non-cellular operators, which will

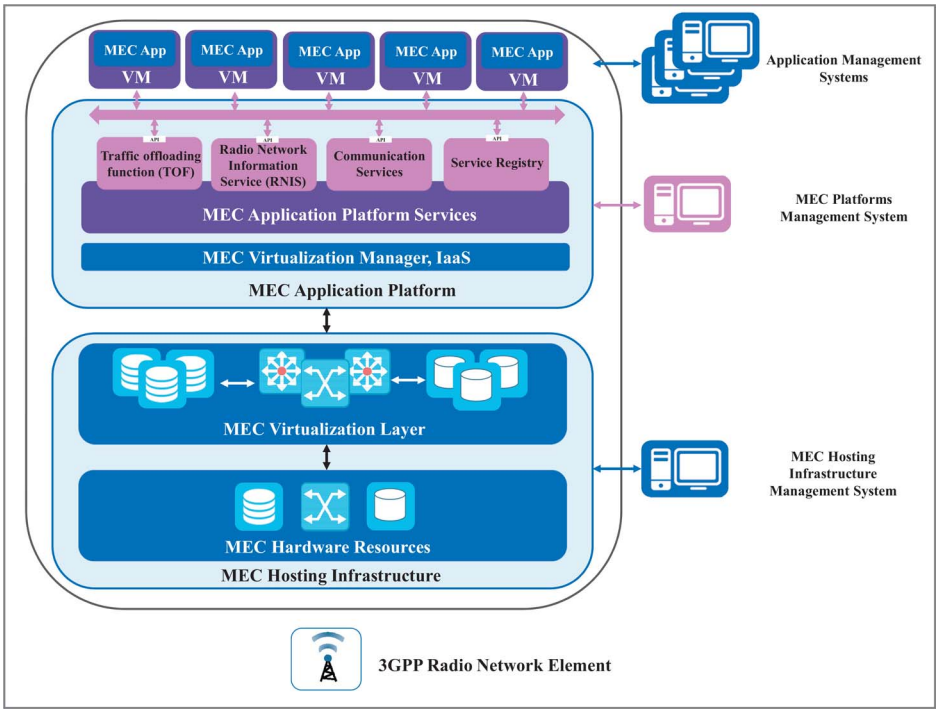


Fig. 13. MEC platform overview [5].

take effects starting from 2017 [230]. Most recently, the *3rd Generation Partnership Project* (3GPP) shows a growing interest in including MEC into its 5G standard, and functionality supports for edge computing have been identified and reported in a recent technical specification document [231]. In this section, we will first introduce the recent standardization efforts from the industry, including the referenced MEC server framework as well as the technical challenges and requirements of MEC systems. Typical use scenarios of MEC will be then elaborated. In addition, we will discuss MEC-related issues in 5G standardizations, including the functionality supports for MEC, and the innovative features in 5G systems with the potential to help realize MEC.

A. Referenced MEC Server Framework

In the MEC introductory technical white paper [5], the ETSI MEC ISG has defined a referenced framework for MEC servers (a.k.a. MEC platforms), where each server consists of a hosting infrastructure and an application platform as shown in Fig. 13. The hosting infrastructure includes the hardware components (such as the computation, memory, and networking resources) and an MEC virtualization layer (which abstracts the detailed hardware implementation to the MEC application platform). Also, the MEC host infrastructure provides the interface to the host infrastructure management system as well as the radio network elements, which, however, are beyond the scope of the MEC initiative due to the availability of multiple implementation options.

The MEC application platform includes an MEC virtualization manager together with an *Infrastructure as a Service (IaaS)* controller, and provides multiple MEC application platform services. The MEC virtualization manager supports a hosting environment by providing IaaS facilities, while the IaaS controller provides a security and resource sandbox (i.e., a virtual environment) for both the applications and MEC platform. The MEC application platform offers four main categories of services, i.e., *traffic offloading function* (TOF), *radio network information services* (RNIS), communication services, and service registry. An MEC application platform management interface is used by the operators for MEC application platform management, supporting the application configuration and life cycle control, as well as VM operation management.

On top of the MEC application platform, the MEC applications are deployed and executed within the VMs, which are managed by their related application management systems and agnostic to the MEC server/platform and other MEC applications.

B. Technical Challenges and Requirements

In this subsection, we will briefly summarize the technical challenges and requirements specified in [5] and [242].

1) Network Integration: As MEC is a new type of service deployed on top of the communication networks, the MEC platform is supposed to be transparent to the 3GPP network architectures, i.e., the existing 3GPP specifications should not be largely affected by the introduction of MEC.

2) Application Portability: Application portability requires MEC applications to be seamlessly loaded and executed by the MEC servers deployed by multiple vendors. This eliminates the need for dedicated development or integration efforts for each MEC platform, and provides more freedom on optimizing the location and execution of MEC applications. It requires the consistency of the MEC application platform management systems, as well as mechanisms used to package, deploy and manage applications from different platforms and vendors.

3) Security: The MEC systems face more security challenges than communication networks due to the integration of computing and IT services. Hence, the security requirements

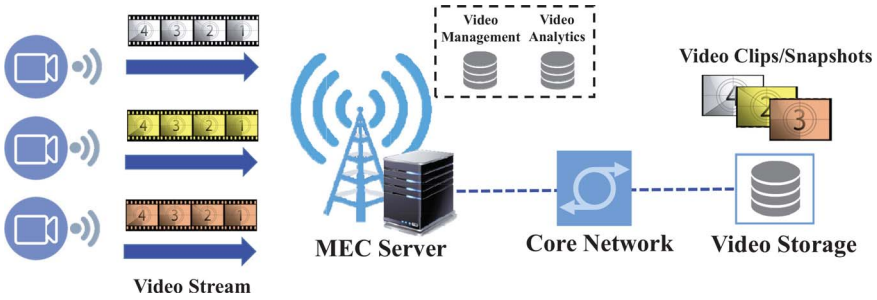


Fig. 14. MEC for video stream analysis [5].

for the 3GPP networks and the IT applications (e.g., isolating different applications as much as possible) should be simultaneously satisfied. Besides, because of the nature of proximity, the physical security of the MEC servers is more vulnerable compared to conventional data centers. Thus, the MEC platforms need to be designed in a way that both logical intrusions and physical intrusions are well protected. Moreover, authorization is an important aspect to prevent the unauthorized/untrusted third-party applications from destroying MEC hosts as well as the valued radio access network.

4) Performance: As mentioned previously, the telecom operators expect that introducing MEC will have minimal impacts on the network performance, e.g., the throughput, latency, and packet loss. Thus, sufficient capacity should be provisioned to process the user traffic in the system deployment stage. Also, because of the highly-virtualized nature, the provided performance may be impaired especially for those applications that require intensive use of hardware resources or have low latency requirements. As a result, how to improve the efficiency of virtualized environments becomes a big challenge.

5) Resilience: The MEC systems should offer certain level of resilience and meet the high-availability requirements demanded by their network operators. The MEC platforms and applications should have fault-tolerant abilities to prevent them from adversely affecting other normal operations of the network.

6) Operation: The virtualization and Cloud technologies make it possible for various parties to participate in the management of MEC systems. Thus, the implementation of the management framework should also consider the diversity of potential deployments.

7) Regulatory and Legal Considerations: The development of MEC systems should meet the regulatory and legal requirements, e.g., the privacy and charging.

Besides the aforementioned challenges and requirements, there still exist more aspects that should be considered in the final MEC standards, such as the support for user mobility, applications/traffic migration, and requirements on the connectivity and storage. However, currently, the standardization efforts and even efforts from the research communities are still on their infant stages.

C. Use Scenarios

MEC will enable numerous mobile applications. In this subsection, we will introduce four typical use scenarios that have been documented by ETSI MEC ISG in [46].

1) Video Stream Analysis Service: Video stream analysis has a broad range of applications such as the vehicular license plate recognition, face recognition, and home security surveillance, for which, the basic operations include object detection and classification. The video analysis algorithms normally have a high computation complexity, and thus it is preferable to move the analysis jobs away from the video-capturing devices (e.g., the camera) to simplify the device design and reduce the cost. If these processing tasks are handled in the central cloud, the video stream should be routed to the core network [232], which will consume a great amount of network bandwidth due to the nature of video stream. By performing the video analysis in the place close to edge devices, the system can not only enjoy the benefits of low latency, but also avoid the problem of network congestion caused by the video stream uploading. The MEC-based video analysis system is shown in Fig. 14, where the edge server should have the ability to conduct video management and analysis, and only the valuable video clips (screenshots) will be backed up to the cloud data centers.

2) Augmented Reality Service: AR is a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory inputs such as sound, video, graphics, or GPS data.[5] Upon analyzing such information, the AR applications can provide additional information in real-time. The AR applications are highly localized and require low latency as well as intensive data processing. One of the most popular applications is the museum video guides, i.e., a handheld mobile device that provides the detailed information of some exhibits that cannot be easily shown on the scene. Online games, such as the Pokémon Go,[6] is another important application that AR techniques play a critical role. An MEC-based AR application system is shown in Fig. 15, where the MEC server should be able to distinguish the requested contents by accurately analyzing the input data, and then transmit the AR data back to the end user. Much attention has been paid on the MEC-enabled AR systems recently, and one demo has been implemented by Intel and roadshowed in the Mobile World Congress 2016 [233].

3) IoT Applications: To simplify the hardware complexity of IoT devices and prolong their battery lives, it is promising to offload the computation-intensive tasks for remote processing and retrieve the results (required action) once the processing is completed. Also, some IoT applications need to obtain

[5]https://en.wikipedia.org/wiki/Augmented_reality
[6]http://www.pokemongo.com/

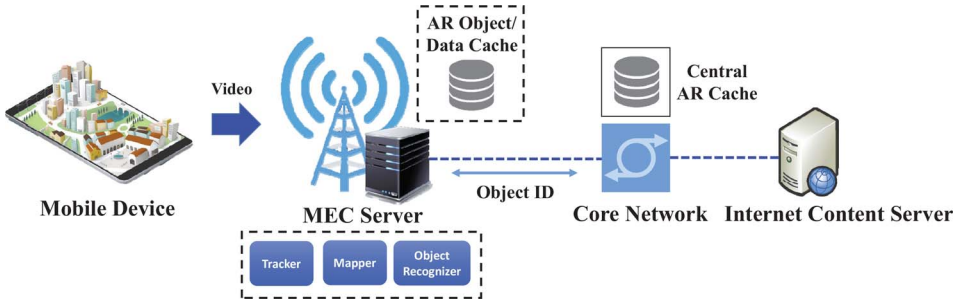


Fig. 15. MEC for AR services [5].

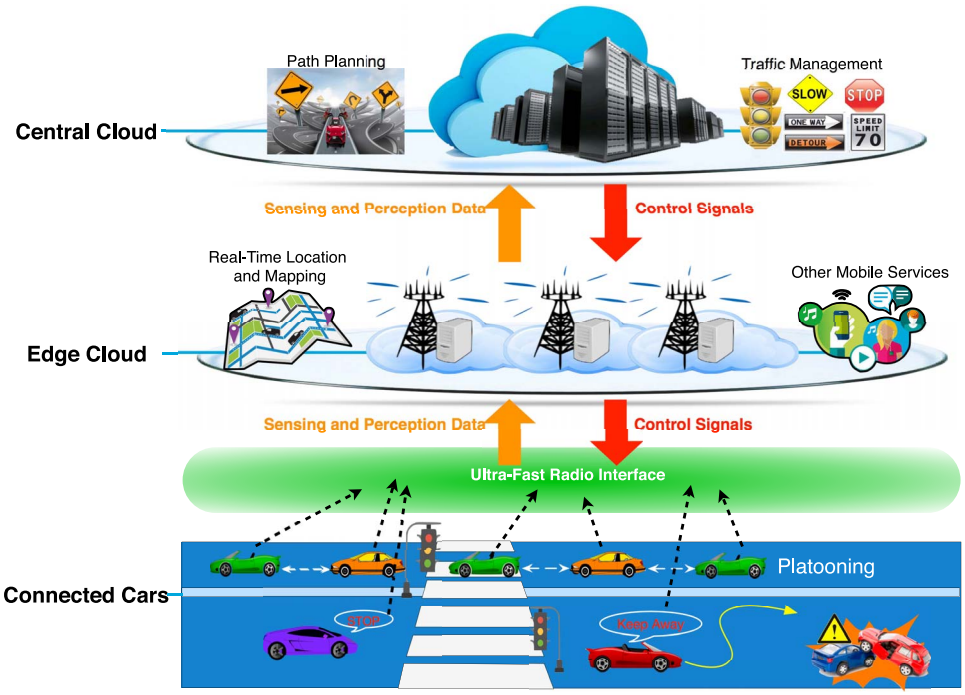


Fig. 16. MEC for connected vehicles.

distributed information for computation, which might be difficult for an IoT device without the aid of an external entity. Since the MEC servers host high-performance computation capabilities and are able to collect distributed information, their deployment will significantly simplify the design of IoT devices, without the need to have strong processing power and capability to receive information from multiple sources for performing meaningful computation. Another important feature of IoT is the heterogeneity of the devices running different forms of protocols, and their management should be accomplished by a low-latency aggregation point (gateway), which could be the MEC server.

4) Connected Vehicles: The connected vehicle technology can enhance safety, reduce traffic congestion, sense vehicles' behaviors, as well as provide opportunities for numerous value-added services such as the car finder and parking location [234]–[236]. However, the maturity of such technology is yet to come as the latency requirement cannot be met with the existing connected car clouds, which contributes to an end-to-end latency between 100ms to 1s. MEC is a key enabling technique for connected vehicles by adding computation and geo-distributed services to roadside BSs. By receiving and analyzing the messages from proximate vehicles and roadside sensors, the connected vehicle cloudlets are able to propagate the hazard warnings and latency-sensitive messages within a 20ms end-to-end delay, allowing the drivers to react immediately (as shown in Fig. 16) and make it possible for autonomous driving. The connected vehicle technology has already attracted extensive attention from the automobile manufacturers (e.g., Volvo, Peugeot), automotive supplier (e.g., BOSCH), telecom operators (e.g., Orange, Vodafone, NTT DOCOMO), telecom vendors (e.g., QualComm, Nokia, Huawei), as well as many research institutes. In November 9 2015, Nokia[7] presented two use cases for connected vehicles on an automotive driving testbed, including the emergency brake light and cooperative passing assistant.

In addition to connected vehicle systems with automobiles, MEC will also be applicable for enabling connected *unmanned aerial vehicles* (UAVs), which play an increasingly important role in various scenarios such as photography, disaster response, inspection and monitoring, precision agriculture, etc. In 2016, Nokia proposed the *UAV traffic management*

[7]https://networks.nokia.com/solutions/mobile-edge-computing

(UTM) based MEC architecture for connected UAVs in [237], where the UTM unit provides functions of fleet management, automated UAV missions, 3D navigation, and collision avoidance. However, as existing mobile networks are mainly designed for users on the ground, UAVs will have very limited connectivity and bandwidth. Therefore, reconfiguring the mobile networks to guarantee the connectivity and low latency between the UAVs and the infrastructure becomes a critical task for designing MEC systems for connected UAVs.

Due to limited space, we omit the description of some other interesting application scenarios, such as active device tracking, RAN-aware content optimization, distributed content and *Domain Name System* (DNS) caching, enterprise networks, as well as safe-and-smart cities. Interested readers may refer to the white papers on MEC [5], [20], [238] for details.

D. MEC in 5G Standardizations

The 5G standard is currently under development, which is to enable the connectivity of a broad range of applications with new functionality, characteristics, and requirements [75]. To achieve these visions, the network features and functionality in 5G networks are foreseen to be migrated from hardware to software, thanks to the recent development of SDN and NFV techniques. Since 2015, MEC (together with SDN and VFN) is recognized by the European *5G Infrastructure Public Private Partnership* (5GPPP) research body as one of the key emerging technologies for 5G networks as it is a natural development in the evolution of mobile BSs and the convergence of IT and telecommunication networking [15]. In April 2017, 3GPP has included *supporting edge computing* as one of the high level features in 5G systems in the technical specification document [231], which will be introduced in this subsection. We have also identified some innovative features of 5G systems, which would pave the way for the realization, standardization and commercialization of MEC.

1) Functionality Supports Offered by 5G Networks: From the 5G network operators' point of view, reducing the end-to-end latency and load on the transport networks are two dominant design targets, which could possibly be achieved with MEC as operators and third part applications could be hosted close to the *user equipment's* (UE's) associated wireless AP. To integrate MEC in 5G systems, the recent 5G technical specifications have explicitly pointed out necessary functionality supports that should be offered by 5G networks for edge computing, as listed below:

- The 5G core network should select the traffic to be routed to the applications in the local data networks.
- The 5G core network selects a *user plane function* (UPF) in proximity to the UE to route and execute the traffic steering from the local data networks via the interface, which should be based on the UE's subscription data, UE location, and the data from the *application function* (AF).
- The 5G network should guarantee the session and service continuity to enable UE and application mobility.
- The 5G core network and AF should provide information to each other via the *network exposure function* (NEF).[8]
- The *policy control function* (PCF)[9] provides rules for QoS control and charging for the traffic routed to the local data network.

2) Innovative Features in 5G to Facilitate MEC: Compared to previous generations of wireless networks, 5G networks possess various innovative features that are beneficial to the realization, standardization, and commercialization of MEC. Three of them will be detailed in this subsection, including the *support service requirement*, *mobility management strategy*, and *capability of network slicing*.

- *Support ServiceRequirement:* In 5G systems, the QoS characteristics (in terms of resource type, priority level, packet delay budget, and packet error rate), which describe the packet forwarding treatment that a QoS flow receives edge-to-edge between the UE and the UPF, are associated with the *5G QoS Indicator* (5QI). In [231], a standardized 5QI to QoS mapping table is provided, showing a broad range of services that can be supported in 5G systems. In particular, 5G systems are able to cater the requirements of latency-sensitive applications (e.g., real-time gaming and *vehicular-to-everything* (V2X) messages, which have a stringent packet budget delay requirement, i.e., $<$ 50ms, and a relatively small packet error rate $< 10^{-3}$), and mission-critical services (e.g., push-to-talk signaling that has both low delay ($<$ 60ms) and small packet error rate ($< 10^{-6}$) requirements). These applications coincide with typical MEC applications as mentioned in Section V-C, i.e., 5G network is a viable choice for wireless communications in MEC systems.
- *Advanced Mobility Management Strategy:* The concept of *mobility pattern* was introduced for designing mobility management strategy for 5G systems. Such strategies may be used by the 5G core network to characterize and optimize UE mobility. Specifically, the mobility pattern could be determined, monitored, and updated by the 5G core network based on the subscription of the UE, statistics of UE mobility, network local policy, and UE assisted information [231]. The mobility pattern not only plays a central role on designing advanced transmission schemes in wireless communication systems, but also becomes a non-negligible design consideration for many MEC applications discussed in Section V-C, e.g., the AR services and connected vehicle applications. Thus, integration of advanced mobility management strategies that make full use of the mobility pattern in 5G network can help to develop an efficient wireless interface for MEC systems. Besides, the mobility pattern obtained from the 5G core network can be further leveraged to design joint radio-and-computational resource management strategies for MEC systems.
- *Capability of Network Slicing:* Network slicing is a form of agile and virtual network architecture that allows multiple network instances to be created on top of a common shared physical infrastructure.[10] Each of

[8]The NEF supports external exposure of capabilities of network functions, which can be categorized into monitoring capability, provisioning capability, and policy/charging capability [231].

[9]The PCF was defined as a stand-alone functional part of the 5G core network that allows to shape the network behaviour based on the operator policies [239].

[10]https://5g.co.uk/guides/what-is-network-slicing/

the network instances is optimized for a specific service, enabling resource isolation and customized network operations [240]. Due to the heterogeneous types of services that 5G systems need to support (different requirements in terms of functionality and performance), network slicing is regarded as an indispensable feature in 5G systems to support different services running across a single radio access network. Existing studies found that network slicing is of supreme need for three use scenarios, including *ultra-reliable and low latency communication* (URLLC), *massive machine-type communication* (mMTC), and *enhanced mobile broadband* (eMBB) [241]. With the capability of network slicing in 5G systems, MEC applications could be provisioned with optimized and dedicated network resources, which could help to reduce the latency incurred by the access networks substantially and support intense access of MEC service subscribers.

VI. Conclusion

MEC is an innovative network paradigm to cater for the unprecedented growth of computation demands and the ever-increasing computation quality of user experience requirements. It aims at enabling Cloud Computing capabilities and IT services in close proximity to end users, by pushing abundant computational and storage resources towards the network edges. The direct interaction between mobile devices and edge servers through wireless communications brings the possibility of supporting applications with ultra-low latency requirement, prolonging device battery lives and facilitating highly-efficient network operations. However, they come along with various new design considerations and unique challenges due to reasons such as the complex wireless environments and the inherent limited computation capacities of MEC servers.

In this survey, we presented a comprehensive overview and research outlook of MEC from the communication perspective. To this end, we first summarized the modeling methodologies on key components of MEC systems such as the computation tasks, communications, as well as computation of mobile devices and MEC servers. This helps to characterize the latency and energy performance of MEC systems. Based upon the system modeling, we conducted a comprehensive literature review on recent research efforts on resource management for MEC under various system architectures, which exploit the concepts of computation offloading, joint radio-and-computational resource allocation, MEC server scheduling, as well as multi-server selection and cooperation. A number of potential research directions were then identified, including MEC deployment issues, cache-enabled MEC, mobility management for MEC, green MEC, as well as security-and-privacy issues in MEC. Key research problems and preliminary solutions for each of these directions were elaborated. Finally, we introduced the recent standardization efforts from industry, along with several typical use scenarios. The comprehensive overview and research outlook on MEC provided in this survey hopefully can serve as useful references and valuable guidelines for further in-depth investigations of MEC.

References

[1] M. Armbrust *et al.*, "Above the clouds: A Berkeley view of cloud computing," Dept. Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, USA, Feb. 2012. [Online]. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[3] N. Wingfield, "Amazon's profits grow more than 800 percent, lifted by cloud services," New York Times, New York, NY, USA, Jul. 2016. [Online]. Available: https://www.nytimes.com/2016/07/29/technology/amazon-earnings-profit.html?_r=0

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[5] "Mobile-edge computing—Introductory technical white paper," White Paper, ETSI, Sophia Antipolis, France, Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[6] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[7] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[8] "Smart wireless devices and the Internet of me," White Paper, Juniper, Sunnyvale, CA, USA, Mar. 2015. [Online]. Available: http://itersnews.com/wp-content/uploads/experts/2015/03/96079Smart-Wireless-Devices-and-the-Internet-of-Me.pdf

[9] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[10] "The Internet of Things: How the next evolution of the Internet is changing everything," White Paper, Cisco, San Jose, CA, USA, Apr. 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM 1st Ed. MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, Aug. 2012, pp. 13–16.

[12] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. ACM Workshop Mobile Big Data*, Hangzhou, China, Jun. 2015, pp. 37–42.

[13] G. I. Klas, "Fog computing and mobile edge cloud gain momentum: Open fog consortium, ETSI MEC and cloudlets," Nov. 2015. [Online]. Available: http://yucianga.info/wp-content/uploads/2015/11/15-11-22-Fog-computing-and-mobile-edge-cloud-gain-momentum-%E2%80%93-Open-Fog-Consortium-ETSI-MEC-Cloudlets-v1.pdf

[14] R. P. Goldberg, "Survey of virtual machine research," *Computer*, vol. 7, no. 9, pp. 34–45, Sep. 1974.

[15] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," White Paper, ETSI, Sophia Antipolis, France, 2015.

[16] C.-Y. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC architectural implications for LTE/LTE-A networks," in *Proc. ACM Workshop Mobility Evol. Internet Archit. (MobiArch)*, New York, NY, USA, Oct. 2016, pp. 13–18.

[17] Z. Q. Jaber and M. I. Younis, "Design and implementation of real time face recognition system (RTFRS)," *Int. J. Comput. Appl.*, vol. 94, no. 12, pp. 15–22, May 2014.

[18] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "Leveraging cloudlets for immersive collaborative applications," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 30–38, Oct./Dec. 2013.

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

[20] "Mobile edge computing use cases & deployment options," White Paper, Juniper, Sunnyvale, CA, USA, Jul. 2016. [Online]. Available: https://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000642-en.pdf

[21] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 52–58, Apr. 2010.

[22] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart., 2014.

[23] M. F. Bari *et al.*, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.

[24] A. Ghiasi and R. Baca, "Overview of largest data centers," presented at the *IEEE 802.3BS Task Force Iterim Meeting*, Norfolk, VA, USA, May 2014. [Online]. Available: http://www.ieee802.org/3/bs/public/14_05/ghiasi_3bs_01b_0514.pdf

[25] S. Clinch, J. Harkes, A. Friday, N. Davies, and M. Satyanarayanan, "How close is close enough? Understanding the role of cloudlets in supporting display appropriation by mobile users," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Lugano, Switzerland, Mar. 2012, pp. 122–127.

[26] 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.

[27] S. Wang *et al.*, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[28] J. Zhang, W. Xie, F. Yang, and Q. Bi, "Mobile edge computing and field trial results for 5G low latency scenario," *China Commun.*, vol. 13, no. 2, pp. 174–182, 2016.

[29] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, San Francisco, CA, USA, Jun. 2010, pp. 49–62.

[30] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[31] "5G automotive vision," White Paper, 5GPPP, Oct. 2015. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf

[32] O. Khalid, M. U. S. Khan, S. U. Khan, and A. Y. Zomaya, "OmniSuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks," *IEEE Trans. Services Comput.*, vol. 7, no. 3, pp. 401–414, Jul./Sep. 2014.

[33] K. Goel and M. Goel, "Cloud computing based e-commerce model," in *Proc. IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. (RTEICT)*, Bengaluru, India, May 2016, pp. 27–30.

[34] G. Riahi, "E-learning systems based on cloud computing: A review," *Proc. Comput. Sci.*, vol. 62, pp. 352–359, Sep. 2015.

[35] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-Health clouds," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 4, pp. 1431–1441, Jul. 2014.

[36] "Understanding 5G: Perspectives on future technological advancements in mobile," GSMA Intell., London, U.K., Dec. 2014. [Online]. Available: https://www.gsmaintelligence.com/research/?file=141208-5g.pdf&download

[37] A. Somov and R. Giaffreda, "Powering IoT devices: Technologies and opportunities," *IEEE IoT Newslett.*, Nov. 2015. [Online]. Available: http://iot.ieee.org/newsletter/november-2015/powering-iot-devices-technologies-and-opportunities.html

[38] R. Kemp *et al.*, "eyeDentify: Multimedia cyber foraging from a smartphone," in *Proc. IEEE Int. Symp. Multimedia*, San Diego, CA, USA, Dec. 2009, pp. 392–399.

[39] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proc. ACM Int. Symp. Multimedia*, Brisbane, QLD, Australia, Oct. 2015, pp. 1271–1274.

[40] W. N. Schilit, "A system architecture for context-aware mobile computing," Ph.D. dissertation, Graduate School Arts Sci., Columbia Univ., New York, NY, USA, 1995.

[41] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[42] S. Nunna *et al.*, "Enabling real-time context-aware collaboration through 5G and mobile edge computing," in *Proc. IEEE Int. Conf. Inf. Technol. New Gener. (ITNG)*, Las Vegas, NV, USA, Apr. 2015, pp. 601–605.

[43] X. Luo, "From augmented reality to augmented computing: A look at cloud-mobile convergence," in *Proc. IEEE Int. Symp. Ubiquitous Virtual Reality*, Gwangju, South Korea, Jul. 2009, pp. 29–32.

[44] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, Mar./Apr. 2011, pp. 267–280.

[45] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Cagliari, Italy, Jul. 2013, pp. 655–659.

[46] "Mobile-edge computing (MEC); Service scenarios," ETSI, Sophia Antipolis, France, Nov. 2015. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf

[47] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[48] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proc. IEEE World Forum Internet Things (WF IoT)*, Milan, Italy, Dec. 2015, pp. 603–608.

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

[50] T. Taleb *et al.*, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[51] H. Liu *et al.*, "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Syst. J.*, to be published.

[52] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. Int. Conf. Adv. Future Internet (AFIN)*, Lisbon, Portugal, Nov. 2014, pp. 48–54.

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

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

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

[56] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, USA, Jun. 2010, pp. 1–7.

[57] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *Proc. IEEE Annu. Consum. Commun. Netw. Conf. (CNCC)*, Las Vegas, NV, USA, Jan. 2017, pp. 160–164.

[58] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time CPU scheduling for mobile multimedia systems," in *Proc. ACM Symp. Oper. Syst. Principles (SOSP)*, Bolton Landing, NY, USA, Oct. 2003, pp. 149–163.

[59] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr./May 2014, pp. 352–357.

[60] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comput.*, to be published.

[61] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge Univ. Press, 2005.

[62] D. Gesbert *et al.*, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[63] S. A. Jafar, "Topological interference management through index coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 529–568, Jan. 2014.

[64] C. Li, J. Zhang, M. Haenggi, and K. B. Letaief, "User-centric intercell interference nulling for downlink small cell networks," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1419–1431, Apr. 2015.

[65] E. Torkildson, U. Madhow, and M. Rodwell, "Indoor millimeter wave MIMO: Feasibility and performance," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4150–4160, Dec. 2011.

[66] X. Yu, J.-C. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 485–500, Apr. 2016.

[67] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[68] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 684–702, Jun. 2003.

[69] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[70] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, Apr. 2012.

[71] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, "Modeling and analysis of K-tier downlink heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 550–560, Apr. 2012.

[72] S. Han, Y.-C. Liang, and B.-H. Soong, "Spectrum refarming: A new paradigm of spectrum sharing for cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1895–1906, May 2016.

[73] Q. Chen, G. Yu, and Z. Ding, "Optimizing unlicensed spectrum sharing for LTE-U and WiFi network coexistence," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2562–2574, Oct. 2016.

[74] P. Kryszkiewicz, A. Kliks, and H. Bogucka, "Small-scale spectrum aggregation and sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2630–2641, Oct. 2016.

[75] "5G radio access—Capabilities and technologies," White Paper, ERICSSON, Stockholm, Sweden, Apr. 2016. [Online]. Available: https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf

[76] T. D. Burd and R. W. Broderson, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, nos. 2–3, pp. 203–221, Aug./Sep. 1996.

[77] W. Yuan and K. Nahrstedt, "Energy-efficient CPU scheduling for multimedia applications," *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 292–331, Aug. 2006.

[78] K. D. Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "The energy/frequency convexity rule: Modeling and experimental validation on mobile devices," in *Proc. Int. Conf. Parallel Process. Appl. Math. (PPAM)*, Warsaw, Poland, Sep. 2013, pp. 793–803.

[79] W. Zhang *et al.*, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[80] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annu. Tech. Conf.*, Boston, MA, USA, Jun. 2010, pp. 1–14.

[81] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[82] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 33, pp. 1397–1411, Mar. 2016.

[83] P. Barham *et al.*, "Xen and the art of virtualization," in *Proc. ACM Symp. Oper. Syst. Principles (SOSP)*, Bolton Landing, NY, USA, Oct. 2003, pp. 164–177.

[84] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Darmstadt, Germany, Jun. 2013, pp. 26–30.

[85] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[86] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[87] S. Vakilinia, M. M. Ali, and D. Qiu, "Modeling of the resource allocation in cloud computing centers," *Comput. Netw.*, vol. 91, pp. 453–470, Nov. 2015.

[88] D. Bruneo, "A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 560–569, Mar. 2014.

[89] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. 34th ACM Annu. Int. Symp. Comput. Archit. (ISCA)*, San Diego, CA, USA, Jun. 2007, pp. 13–23.

[90] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-efficient virtual machine provision algorithms for cloud systems," in *Proc. IEEE Utility Cloud Comput. (UCC)*, Melbourne, VIC, Australia, Dec. 2011, pp. 81–88.

[91] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.

[92] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[93] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.

[94] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[95] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[96] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr./May 2015, pp. 1894–1902.

[97] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan. 2015.

[98] S. Khalili and O. Simeone, "Inter-layer per-mobile optimization of cloud mobile computing: A message-passing approach," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 6, pp. 814–827, Jun. 2016.

[99] P. D. Lorenzo, S. Barbarossa, and S. Sardellitti, "Joint optimization of radio resources and code partitioning in mobile edge computing." [Online]. Available: http://arxiv.org/abs/1307.3835v3

[100] S. E. Mahmoodi, K. P. Subbalakshmi, and V. Sagar, "Cloud offloading for multi-radio enabled mobile devices," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 5473–5478.

[101] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[102] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[103] S. Chen, Y. Wang, and M. Pedram, "A semi-Markovian decision process based control method for offloading tasks from mobile devices to the cloud," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013, pp. 2885–2890.

[104] S.-T. Hong and H. Kim, "QoE-aware computation offloading scheduling to capture energy-latency tradeoff in mobile clouds," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, London, U.K., Jun. 2016, pp. 1–9.

[105] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[106] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306–2316, 2015.

[107] D. T. Hoang, D. Niyato, and P. Wang, "Optimal admission control policy for mobile cloud computing hotspot with cloudlet," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Paris, France, Apr. 2012, pp. 3145–3149.

[108] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[109] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, to be published.

[110] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading." [Online]. Available: https://arxiv.org/pdf/1704.00163.pdf

[111] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[112] M.-H. Chen, B. Liang, and D. Ming, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, Apr. 2017, pp. 1863–1871.

[113] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2016.

[114] M.-H. Chen, M. Dong, and B. Liang, "Multi-user mobile cloud offloading game with computing access point," in *Proc. IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, Oct. 2016, pp. 64–69.

[115] X. Ma, C. Lin, X. Xiang, and C. Chen, "Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing," in *Proc. ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst. (MSWiM)*, Cancún, Mexico, Nov. 2015, pp. 271–278.

[116] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

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

[118] M. Molina, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint scheduling of communication and computation resources in multiuser wireless application offloading," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Washington, DC, USA, Sep. 2014, pp. 1093–1098.

[119] Y. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and CPU time allocation for mobile edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[120] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.

[121] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2239–2244.

[122] M. Jo, T. Maksymyuk, B. Strykhalyuk, and C.-H. Cho, "Device-to-device-based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 50–58, Jun. 2015.

[123] Z. Sheng, C. Mahapatra, V. Leung, M. Chen, and P. Sahu, "Energy efficient cooperative computing in mobile wireless sensor networks," *IEEE Trans. Cloud Comput.*, to be published.

[124] J. Song, Y. Cui, M. Li, J. Qiu, and R. Buyya, "Energy-traffic tradeoff cooperative offloading for mobile cloud computing," in *Proc. IEEE/ACM Int. Symp. Qual. Service (IWQoS)*, Hong Kong, May 2014, pp. 284–289.

[125] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing." [Online]. Available: https://arxiv.org/pdf/1704.06777.pdf

[126] C. You and K. Huang, "Exploiting non-causal CPU-state information for energy-efficient mobile cooperative computing." [Online]. Available: https://arxiv.org/pdf/1704.04595.pdf

[127] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks." [Online]. Available: https://arxiv.org/pdf/1703.06058.pdf

[128] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 34–44, Jun. 2013.

[129] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing," in *Proc. IEEE Glob. Commun. Conf. Workshops (GC WKSHPS)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[130] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Redondo Beach, CA, USA, Jul./Aug. 2012, pp. 279–284.

[131] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[132] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[133] R. Yu *et al.*, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[134] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Oulu, Finland, Jun. 2017, pp. 1–6.

[135] S. Wang *et al.*, "Mobility-induced service migration in mobile micro-clouds," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Baltimore, MD, USA, Oct. 2014, pp. 835–840.

[136] R. Urgaonkar *et al.*, "Dynamic service migration and workload scheduling in edge-clouds," *Perform. Eval.*, vol. 91, pp. 205–228, Sep. 2015.

[137] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Accoust. Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 3516–3520.

[138] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[139] S. Wang and S. Dey, "Modeling and characterizing user experience in a cloud server based mobile gaming approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1–7.

[140] 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.

[141] Y. Shi, J. Zhang, B. O'Donoghue, and K. B. Letaief, "Large-scale convex optimization for dense wireless cooperative networks," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4729–4743, Sep. 2013.

[142] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 77–83, Oct. 2016.

[143] T. H. Luan *et al.*, "Fog computing: Focusing on mobile users at the edge." [Online]. Available: https://arxiv.org/pdf/1502.01815v3.pdf

[144] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[145] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, "An approach to ad hoc cloud computing." [Online]. Available: https://arxiv.org/pdf/1002.4738v1.pdf

[146] T. Truong-Huu, C.-K. Tham, and D. Niyato, "A stochastic workload distribution approach for an ad hoc mobile cloud," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Singapore, Dec. 2014, pp. 174–181.

[147] D. M. Shila, W. Shen, Y. Cheng, X. Tian, and X. S. Shen, "AMCloud: Toward a secure autonomic mobile ad hoc cloud computing system," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 74–81, Apr. 2017.

[148] X. Hou *et al.*, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[149] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, Sep. 2009.

[150] J. G. Andrews, F. Baccelli, and R. K. Ganti, "A tractable approach to coverage and rate in cellular networks," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 3122–3134, Nov. 2011.

[151] M. Haenggi, *Stochastic Geometry for Wireless Networks*. New York, NY, USA: Cambridge Univ. Press, 2012.

[152] C. Li, J. Zhang, J. G. Andrews, and K. B. Letaief, "Success probability and area spectral efficiency in multiuser MIMO HetNets," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1544–1556, Apr. 2016.

[153] N. Vastardis and K. Yang, "An enhanced community-based mobility model for distributed mobile social networks," *J. Ambient Intell. Humanized Comput.*, vol. 5, no. 1, pp. 65–75, Feb. 2014.

[154] "Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020," White Paper, Cisco, San Jose, CA, USA, 2016. [Online]. Available: https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf

[155] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[156] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[157] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 1107–1115.

[158] A. S. Gomes *et al.*, "Edge caching with mobility prediction in virtualized LTE mobile networks," *Future Gener. Comput. Syst.*, vol. 70, pp. 148–162, May 2017.

[159] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[160] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers," *Future Gener. Comput. Syst.*, vol. 28, no. 2, pp. 358–367, Feb. 2012.

[161] B. Li *et al.*, "EnaCloud: An energy-saving application live placement approach for cloud computing environments," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Bengaluru, India, Sep. 2009, pp. 17–24.

[162] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, Dec. 2013.

[163] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1431–1441, Aug. 2013.

[164] H. Rheingold, *Virtual Reality: Exploring the Brave New Technologies*. New York, NY, USA: Simon & Schuster Adult, 1991.

[165] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. IEEE/IFIP Annu. Conf. Wireless Demand Netw. Syst. Services (WONS)*, Jackson, WY, USA, Feb. 2017, pp. 165–172.

[166] E. Baştuğ, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–11, Feb. 2015.

[167] Y. Cui, Y. Wu, and D. Jiang, "Analysis and optimization of caching and multicasting in large-scale cache-enabled information-centric networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–7.

[168] V. Suryaprakash, J. Møller, and G. Fettweis, "On the modeling and analysis of heterogeneous radio access networks using a Poisson cluster process," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 1035–1047, Feb. 2015.

[169] D. Lopez-Perez, I. Guvenc, and X. Chu, "Mobility management challenges in 3GPP heterogeneous networks," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 70–78, Dec. 2012.

[170] A. Damnjanovic *et al.*, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[171] M. Kassar, B. Kervella, and G. Pujolle, "An overview of vertical handover decision strategies in heterogeneous wireless networks," *Comput. Commun.*, vol. 31, no. 10, pp. 2607–2620, Jun. 2008.

[172] C. Wang, Y. Li, and D. Jin, "Mobility-assisted opportunistic computation offloading," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1779–1782, Oct. 2014.

[173] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[174] K. Lee and I. Shin, "User mobility model based computation offloading decision for mobile cloud," *J. Comput. Sci. Eng.*, vol. 9, no. 3, pp. 155–162, Sep. 2015.

[175] M. R. Rahimi, N. Venkatasubramanian, and A. V. Vasilakos, "MuSIC: Mobility-aware optimal service allocation in mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Santa Clara, CA, USA, Jun. 2013, pp. 75–82.

[176] A. Prasad, P. Lundén, M. Moisio, M. A. Uusitalo, and Z. Li, "Efficient mobility and traffic management for delay tolerant cloud data in 5G networks," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Hong Kong, Aug./Sep. 2015, pp. 1740–1745.

[177] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[178] S.-W. Ko, K. Huang, S.-L. Kim, and H. Chae, "Live prefetching for mobile computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3057–3071, May 2017.

[179] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[180] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, Jan./Mar. 2015.

[181] C.-A. Chen, R. Stoleru, and G. G. Xie, "Energy-efficient and fault-tolerant mobile cloud storage," in *Proc. IEEE Int. Conf. Cloud Netw. (CloudNet)*, Pisa, Italy, Oct. 2016, pp. 51–57.

[182] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Gener. Comput. Syst.*, vol. 70, pp. 138–147, May 2017.

[183] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Services Comput.*, vol. 5, no. 2, pp. 164–177, Apr./Jun. 2012.

[184] Y. Zhang, J. Yan, and X. Fu, "Reservation-based resource scheduling and code partition in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, Apr. 2016, pp. 962–967.

[185] X. Jin, F. Zhang, A. V. Vasilakos, and Z. Liu, "Green data centers: A survey, perspectives, and future directions." [Online]. Available: https://arxiv.org/pdf/1608.00687.pdf

[186] X. Sun and N. Ansari, "Green cloudlet network: A distributed green mobile cloud network," *IEEE Netw.*, vol. 31, no. 1, pp. 64–70, Jan./Feb. 2017.

[187] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[188] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[189] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *Proc. IEEE Int. Green Comput. Conf. (IGCC)*, San Jose, CA, USA, Jun. 2012, pp. 1–10.

[190] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1743–1753, Jun. 2015.

[191] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. IEEE/ACM Int. Conf. Cluster Cloud Grid Comput. (CCGrid)*, Melbourne, VIC, Australia, May 2010, pp. 826–831.

[192] X. Li, J. Wu, S. Tang, and S. Lu, "Let's stay together: Towards traffic aware virtual machine placement in data centers," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 1842–1850.

[193] L. Chen and H. Shen, "Consolidating complementary VMs with spatial/temporal-awareness in cloud datacenters," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 1033–1041.

[194] Z. Han *et al.*, "Dynamic virtual machine management via approximate Markov decision process," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[195] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[196] S. Ulukus *et al.*, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[197] Y. Mao, Y. Luo, J. Zhang, and K. B. Letaief, "Energy harvesting small cell networks: Feasibility, deployment, and operation," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 94–101, Jun. 2015.

[198] "Battery life concerns mobile users," CNN, Atlanta, GA, USA, Sep. 2005. [Online]. Available: http://edition.cnn.com/2005/TECH/ptech/09/22/phone.study/

[199] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[200] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[201] C. Chen, B. He, and X. Tang, "Green-aware workload scheduling in geographically distributed data centers," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Taipei, Taiwan, Nov. 2012, pp. 82–89.

[202] C. Dong, F. Kong, X. Liu, and H. Zeng, "Green power analysis for geographical load balancing based datacenters," in *Proc. IEEE Int. Green Comput. Conf. (IGCC)*, Arlington, VA, USA, Jun. 2013, pp. 1–8.

[203] X. Sun, N. Ansari, and Q. Fan, "Green energy aware avatar migration strategy in green cloudlet networks," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Vancouver, BC, Canada, Nov. 2015, pp. 139–146.

[204] T. Chen, Y. Zhang, X. Wang, and G. B. Giannakis, "Robust workload and energy management for sustainable data centers," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 651–664, Mar. 2016.

[205] Y. Luo, J. Zhang, and K. B. Letaief, "Transmit power minimization for wireless networks with energy harvesting relays," *IEEE Trans. Commun.*, vol. 64, no. 3, pp. 987–1000, Mar. 2016.

[206] J. Gong, S. Zhou, and Z. Niu, "Optimal power allocation for energy harvesting and power grid coexisting wireless communication systems," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 3040–3049, Jul. 2013.

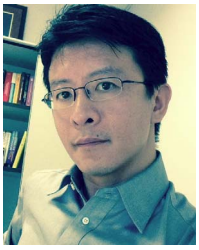
[207] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013.
[208] Y. Mao, J. Zhang, and K. B. Letaief, "Grid energy consumption and QoS tradeoff in hybrid energy supply wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3573–3586, May 2016.
[209] W. C. Brown, "The history of power transmission by radio waves," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-32, no. 9, pp. 1230–1242, Sep. 1984.
[210] H. Ju and R. Zhang, "Throughput maximization in wireless powered communication networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 418–428, Jan. 2014.
[211] K. Huang and V. K. N. Lau, "Enabling wireless power transfer in cellular networks: Architecture, modeling and deployment," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 902–912, Feb. 2014.
[212] Z. Chang *et al.*, "Energy efficient resource allocation for wireless power transfer enabled collaborative mobile clouds," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3438–3450, Dec. 2016.
[213] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, to be published.
[214] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms Syst. Appl. (WASA)*, Qufu, China, Aug. 2015, pp. 1–10.
[215] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 675–685, Dec. 2011.
[216] A. M. Y. Ahmed and D. Qian, "An optimization of security and trust management in distributed systems," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, Ghaziabad, India, Feb. 2013, pp. 120–126.
[217] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 6, pp. 568–581, Nov./Dec. 2014.
[218] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, "Attribute-based authenticated key exchange," in *Proc. Aust. Conf. Inf. Security Privacy (ACISP)*, Sydney, NSW, Australia, Jul. 2010, pp. 1–25.
[219] H. M. Pimentel, S. Kopp, M. A. Simplicio, Jr., R. M. Silveira, and G. Bressan, "OCP: A protocol for secure communication in federated content networks," *Comput. Commun.*, vol. 68, pp. 47–60, Sep. 2015.
[220] M. Liyanage, A. B. Abro, M. Ylianttila, and A. Gurtov, "Opportunities and challenges of software-defined mobile networks in network security," *IEEE Security Privacy*, vol. 14, no. 4, pp. 34–44, Jul./Aug. 2016.
[221] W. Yang and C. Fung, "A survey on security in network functions virtualization," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Seoul, South Korea, Jun. 2016, pp. 15–19.
[222] B. Liang, "Mobile edge computing," in *Key Technologies for 5G Wireless Systems*, V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2017. [Online]. Available: http://www.comm.utoronto.ca/∼liang/publications/Chapter_MEC_2016.pdf
[223] "Providing security in NFV: Challenges and opportunities," White Paper, Alcatel-Lucent, Colombes, France, 2014. [Online]. Available: http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2014/10172-providing-security-nfv.pdf
[224] C. Wang, K. Ren, and J. Wang, "Secure optimization computation outsourcing in cloud computing: A case study of linear programming," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 216–229, Jan. 2016.
[225] R. Gennaro, C. Gentry, and P. Bryan, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. Annu. Conf. Adv. Cryptol.*, Santa Barbara, CA, USA, Aug. 2010, pp. 465–482.
[226] "Executive briefing—Mobile edge computing (MEC) initiative," ETSI, Sophia Antipolis, France, Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mec%20executive%20brief%20v1%2028-09-14.pdf
[227] "Mobile edge computing (MEC): Terminology," ETSI, Sophia Antipolis, France, Mar. 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/01.01.01_60/gs_MEC001v010101p.pdf
[228] "Mobile edge computing (MEC); Framework and reference architecture," ETSI, Sophia Antipolis, France, Mar. 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf
[229] "ETSI first mobile edge computing proof of concepts at MEC world congress," ETSI, Sophia Antipolis, France, Sep. 2016.[Online]. Available: http://www.etsi.org/news-events/news/1119-2016-09-news-etsifirst-mobile-edge-computing-proof-of-concepts-at-mec-world-congress
[230] N. Sprecher, J. Friis, R. Dolby, and J. Reister, "Edge computing prepares for a multi-access future," Sep. 2016. [Online]. Available: http://www.telecomtv.com/articles/mec/edge-computing-prepares-for-a-multi-access-future-13986/
[231] "Technical specification group services and system aspects; System architecture for the 5G systems; Stage 2 (Release 15)," 3GPP Standard TS 23.501 V0.4.0, Apr. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144
[232] A. Anjum, T. Abdullah, M. Tariq, Y. Baltaci, and N. Antonopoulos, "Video stream analysis in clouds: An object detection and classification framework for high performance video analytics," *IEEE Trans. Cloud Comput.*, to be published.
[233] "Intel mobile edge computing technology improves the augmented reality experience," Intel, Santa Clara, CA, USA, Sep. 2016. [Online]. Available: https://www.youtube.com/watch?v=ZSkWJYeKjnk
[234] P. Papadimitratos, A. D. L. Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 84–95, Nov. 2009.
[235] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.
[236] E. Uhlemann, "Introducing connected vehicles," *IEEE Veh. Technol. Mag.*, vol. 10, no. 1, pp. 23–31, Mar. 2015.
[237] "UTM infrastructure and connected society," NOKIA, Espoo, Finland, 2016. [Online]. Available: https://rpas-civops.com/wp-content/uploads/2016/11/S7.2_Nokia_DE_V1.pdf
[238] "Using mobile edge computing to improve mobile network performance and profitability," Saguna, Nashua, NH, USA, and Intel, Santa Clara, CA, USA, 2016. [Online]. Available: https://networkbuilders.intel.com/docs/Saguna-and-Intel-Using-Mobile-Edge-Computing-to-Improve-Mobile-Network-Performance-and-Profitability.pdf
[239] "Policy control function in 5G," ERICSSON, Stockholm, Sweden, Jan. 2017. [Online]. Available: http://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=756820
[240] "Description of network slicing concept," NGMN Alliance, San Diego, CA, USA, Jan. 2016. [Online]. Available: https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf
[241] A. Nakao, "Network softwarization and slicing: Ongoing developments in standard developing organizations," presented at the Keynote *IEEE Conf. Stand. Commun. Netw. (CSCN)* Berlin, Germany, Oct./Nov. 2016. [Online]. Available: http://cscn2016.ieee-cscn.org/document.pdf
[242] "Mobile edge computing (MEC); Technical requirements," ETSI, Sophia Antipolis, France, Mar. 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf

Yuyi Mao (S'14) received the B.Eng. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2013. He is currently pursuing the Ph.D. degree in electronic and computer engineering with the Hong Kong University of Science and Technology, Hong Kong. His current research interests include cooperative communications, energy harvesting communications, green cellular networks with hybrid energy supplies, mobile-edge computing, and stochastic optimization.

Changsheng You (S'15) received the B.S. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2014. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with the University of Hong Kong, Hong Kong. His research interests include mobile-edge computing, fog computing, wireless power transfer, energy harvesting system, and convex optimization.

Jun Zhang (S'06–M'10–SM'15) received the B.Eng. degree in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2004, the M.Phil. degree in information engineering from the Chinese University of Hong Kong, Hong Kong, in 2006, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2009. He is currently a Research Assistant Professor with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests include dense wireless cooperative networks, mobile edge caching and computing, cloud computing, and big data analytics systems.

Dr. Zhang has co-authored the book entitled *Fundamentals of LTE* (Prentice-Hall, 2010). He was a recipient of the 2016 Young Author Best Paper Award of the IEEE Signal Processing Society and the 2016 IEEE ComSoc Asia–Pacific Best Young Researcher Award. He was a co-recipient of the 2016 Marconi Prize Paper Award in Wireless Communications, the 2014 Best Paper Award for the *EURASIP Journal on Advances in Signal Processing*, an IEEE ICC Best Paper Award in 2016, and an IEEE PIMRC Best Paper Award in 2014. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and is a Guest Editor of the Special Section on Mobile Edge Computing for Wireless Networks in the IEEE ACCESS. He frequently serves on the technical program committees of major IEEE conferences in wireless communications, such as ICC, GLOBECOM, WCNC, and VTC, and served as a MAC Track Co-Chair for IEEE WCNC 2011.



Kaibin Huang (M'08–SM'13) received the B.Eng. (First Class Hons.) and M.Eng. degrees from the National University of Singapore, and the Ph.D. degree from the University of Texas at Austin (UT Austin), all in electrical engineering.

Since 2014, he has been an Assistant Professor with the Department of Electrical and Electronic Engineering (EEE), University of Hong Kong. He is an Adjunct Professor with the School of EEE, Yonsei University, South Korea. He was a Faculty Member with the Department of Applied Mathematics (AMA), Hong Kong Polytechnic University and the Department of EEE, Yonsei University. He had been a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, Hong Kong University of Science and Technology from 2008 to 2009 and an Associate Scientist with the Institute for Infocomm Research, Singapore, from 1999 to 2004. His research interests focus on the analysis and design of wireless networks using stochastic geometry and multiantenna techniques.

Dr. Huang was a recipient of the 2015 IEEE ComSoc Asia–Pacific Outstanding Paper Award, the Outstanding Teaching Award from Yonsei, Motorola Partnerships in Research Grant, the University Continuing Fellowship from UT Austin, and the Best Paper Award from IEEE GLOBECOM 2006 and PolyU AMA in 2013. He frequently serves on the technical program committees of major IEEE conferences in wireless communications. He has been the Technical Chair/Co-Chair for the IEEE CTW 2013, the Wireless Communications Symposium of IEEE GLOBECOM 2017, the Communication Theory Symposium of IEEE GLOBECOM 2014, and the Advanced Topics in Wireless Communications Symposium of IEEE/CIC ICCC 2014. He has been the Track Chair/Co-Chair for IEEE PIMRC 2015, IEE VTC Spring 2013, Asilomar 2011, and IEEE WCNC 2011. He is currently an Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Series on Green Communications and Networking, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE WIRELESS COMMUNICATIONS LETTERS. He was also a Guest Editor for the JSAC Special Issues on Communications Powered by Energy Harvesting and an Editor for *IEEE/KICS Journal of Communications and Networks* from 2009 to 2015. He is an elected member of the SPCOM Technical Committee of the IEEE Signal Processing Society.

Khaled B. Letaief (S'85–M'86–SM'97–F'03) received the B.S. (with Distinction), M.S., and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1984, 1986, and 1990, respectively.

From 1990 to 1993, he was a Faculty Member with the University of Melbourne, Melbourne, Australia. He has been with the Hong Kong University of Science and Technology (HKUST), Hong Kong, where he is known as one of HKUST's most distinguished professors. He has held numerous administrative positions, including the Head of the Department of Electronic and Computer Engineering, the Director of Huawei Innovation Laboratory, and the Director of the Hong Kong Telecom Institute of Information Technology. He has also served as the Chair Professor and the Dean of Engineering. Under his leadership, the School of Engineering has dazzled in international rankings (rising from 26 in 2009 to 14 in the world in 2015, according to QS World University Rankings). In 2015, he joined Hamad bin Khalifa University, Doha, Qatar, as a Provost to help establish a research-intensive university in Qatar in partnership with esteemed universities that include Northwestern University, Carnegie Mellon University, Cornell University, and Texas A&M University.

Dr. Letaief is an internationally recognized leader in wireless communications with research interests in green communications, Internet of Things, Cloud-RANs, and 5G systems. He has over 560 journal and conference papers in the above areas, and given keynote talks as well as courses all over the world. He also has 15 patents, including 11 U.S. patents

He served as a consultant for different organizations, including Huawei, ASTRI, ZTE, Nortel, PricewaterhouseCoopers, and Motorola. He is the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and has served on the editorial board of other prestigious journals. He has been a dedicated teacher committed to excellence in teaching and scholarship. He was a recipient of the Michael G. Gale Medal for Distinguished Teaching (highest university-wide teaching award and only one recipient/year is honored for his/her contributions).

He was a recipient of many other distinguished awards, including the 2007 IEEE Joseph LoCicero Publications Exemplary Award, the 2009 IEEE Marconi Prize Award in Wireless Communications, the 2010 Purdue University Outstanding Electrical and Computer Engineer Award, the 2011 IEEE Harold Sobol Award, the 2016 IEEE Marconi Prize Award in Wireless Communications, and over 11 IEEE best paper awards.

He is recognized as a long-time volunteer with dedicated service to professional societies and in particular IEEE, where he has served in many leadership positions. These include treasurer, the vice-president for conferences and the Vice-President for Technical Activities of IEEE Communications Society.

He is a fellow of HKIE. He was also a recipient of the ISI Highly Cited Researcher Award by Thomson Reuters (which places him among the top 250 preeminent individual researchers in the field of computer science and engineering) and will be the President of IEEE Communications Society in 2018.