A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet

JU REN, DEYU ZHANG, SHIWEN HE, and YAOXUE ZHANG, Central South University, China
TAO LI, University of Florida, U.S.A.

Sending data to the cloud for analysis was a prominent trend during the past decades, driving cloud computing as a dominant computing paradigm. However, the dramatically increasing number of devices and data traffic in the Internet-of-Things (IoT) era are posing significant burdens on the capacity-limited Internet and uncontrollable service delay. It becomes difficult to meet the delay-sensitive and context-aware service requirements of IoT applications by using cloud computing alone. Facing these challenges, computing paradigms are shifting from the centralized cloud computing to distributed edge computing. Several new computing paradigms, including Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet, have emerged to leverage the distributed resources at network edge to provide timely and context-aware services. By integrating end devices, edge servers, and cloud, they form a hierarchical IoT architecture, i.e., End-Edge-Cloud orchestrated architecture to improve the performance of IoT systems. This article presents a comprehensive survey of these emerging computing paradigms from the perspective of end-edge-cloud orchestration. Specifically, we first introduce and compare the architectures and characteristics of different computing paradigms. Then, a comprehensive survey is presented to discuss state-of-the-art research in terms of computation offloading, caching, security, and privacy. Finally, some potential research directions are envisioned for fostering continuous research efforts.

CCS Concepts: • **Human-centered computing** → *Ubiquitous computing*; • **Computer systems organization** → *Cloud computing*;

Additional Key Words and Phrases: End-edge-cloud orchestration, network computing, transparent computing, mobile edge computing, fog computing, cloudlet

ACM Reference format:
Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, and Tao Li. 2019. A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet. *ACM Comput. Surv.* 52, 6, Article 125 (October 2019), 36 pages.
https://doi.org/10.1145/3362031

This article is supported by the National Natural Science Foundation of China (No. 61702562 and 61702561), Huxiang Young Talents Plan of Hunan Province, 111 Project (No. B18059), International Science & Technology Cooperation Program of China (No. 2013DFB10070), and China Hunan Provincial Science & Technology Program (No. 2012GK4106).

Authors' addresses: J. Ren, D. Zhang (corresponding author), S. He (corresponding author), and Y. Zhang, School of Computer Science and Engineering, Central South University, Changsha 410083, China; emails: {renju, zdy876, shiwen.he.cn, zyx}@csu.edu.cn; T. Li, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611; email: taoli@ece.ufl.edu.

0360-0300/2019/10-ART125 $15.00
https://doi.org/10.1145/3362031

1 INTRODUCTION

In the past few decades, the development of Internet and wireless communication technologies has provided a very convenient channel for information exchange in people's daily life. By 2019, the number of global mobile terminals increases exponentially to about 2.8 billion. Especially with the advance of artificial intelligence and intelligent science, the number of intelligent lightweight devices has increased exponentially, and the interconnection of all things has become the main trend of the development of wireless communication networks and the Internet [125]. It also implies the coming of the era of Internet of Things (IoT) with a large number of sensors, actuators, and mobile devices deployed at the network edge. A report from Cisco shows that the monthly global mobile data traffic will be 49 exabytes by 2021 with a compound annual growth rate of 47% from 2016 to 2021 [7]. A considerable part of the computing tasks generated by these devices, such as virtual reality, augmented reality, and industrial control, require timely and context-aware processing. As a result, processing massive data traffic is a key feature of the future Internet and wireless communication systems. Furthermore, high data rate and low delivery latency become two key performance indices of the future Internet and wireless communication networks. It implies that powerful computation devices need to process massive data traffic, and high data rate transmission links are also necessary to transfer the data traffic for the Internet and wireless communication networks, respectively.

From the perspective of wireless communication systems, ultra-dense networks, massive multiple-input multiple-output (MIMO), and high-frequency communications have been regarded as promising ways to meet the growing demands of future wireless communications, such as 5G wireless systems. Compared to 4G, 5G is envisioned to receive a 1,000× capacity increase by leveraging these technologies. Besides the dramatic capacity enhancement, it is also anticipated to achieve significant improvement in data transmission rate, network reliability, spectral and energy efficiency, and so on [133]. It also implies that the future wireless communication technologies designed in 5G systems provide powerful capabilities to convey the data traffic generated by various communication devices.

However, in the past few decades, various computing architectures and paradigms are designed to provide powerful capabilities of processing data traffics from the view point of the Internet. Since the advent of the first computer, ENIAC, the development of the computer, the Internet, and information technologies has brought people into an era of information explosion. To further meet the various requirements of the information society, there is a need for revolutionary changes in computer networks, computing modes, storage modes, and application modes. The development of computing modes has gone through the stages of single computer computing, cluster computing, network computing, and cloud computing. The appearance of cluster computing is to address the shortcoming of single computer computing, which cannot process gigantic computing data services. More flexible network computing is developed to increase the business processing capabilities of cluster computing in terms of heterogeneity, dynamics, distribution, and scalability. Although network computing can provide considerable computation power to process data, it still cannot satisfy the ever-increasing demands caused by the exponentially growing mobile devices and data traffic.

Cloud computing with centralized computing and storage resources, which has been regarded as the second generation of network computing and considered as one of the most promising technologies in 21th century, provides powerful capabilities of computation and storage to address the computing challenges. In particular, cloud computing can provide elastic services and data-intensive analysis for end-users over a wide area network (WAN). Therefore, users can be empowered with seemingly unlimited resources without building new computing infrastructures. The global revenue brought by cloud computing is forecasted by Gartner to grow from

$209.2 billions to $246.8 billions [8]. With the remarkable economic benefits of cloud computing, it is likely to stay firmly on the computing landscape [129]. However, although the centralization of computing resource in cloud facilitates resource management and maintenance, there are diffculties for cloud computing to satisfy the service demands of the new trend of delay-sensitive applications in the IoT era. The first issue is the unacceptable WAN latency, which is unlikely to be improved in the foreseeable future, since the design objective of WAN mainly focuses on improving the efficiency of bandwidth and links [130]. The second issue is that the traffic capacity of WAN will be significantly challenged by the dramatically increasing amount of data generated by IoT devices. For example, in an airport surveillance application, several thousand video cameras are deployed for security purposes, each of which produces data at 12 Mbps [10]. To analyze solely video data at the central cloud server, hundreds of Gbps bandwidth is required to collect the video data, which far exceeds the traffic capacity of current WANs. Last, cloud computing has intrinsic disadvantages of supporting context-aware computing for IoT applications, since it works in a remote and centralized computing way.

To address these issues, several new network computing paradigms have emerged to offer computing resources in the proximity of end-users. In such a way, delay-sensitive and context-aware services can be offered without the involvement of WAN [90]. Emerging network computing paradigms, including transparent computing (TC), fog computing (Fog), mobile edge computing (MEC), and cloudlet, have attracted extensive attention in industry and academia. These paradigms employ small-scale edge servers with limited computation resources to timely serve end-users at the network edge. The edge servers can either be temporary devices such as smart phones, laptops, advanced routers, and micro servers, and so on, and also can be some nearby infrastructures. Fog, MEC, and cloudlet can be deemed as extending cloud services to the network edge, since they exploit the similar computation offloading and storage management schemes. However, in the vision of TC, computing and storage are separated into end-devices and remote servers [176]. Specifically, TC encourages end-devices and their nearby devices to undertake the computing tasks and fetch the software and data from remote servers. As a result, the computing capabilities of modern devices can be fully exploited. It is notable that all the above-mentioned computing paradigms emphasize serving end-users at the edge and to serve the delay-sensitive applications in the IoT context.

Motivated by the advances of the emerging computing paradigms, we are likely to see a hierarchical computing architecture that can revolutionize the current cloud computing architecture [50]. It consists of large-scale central servers, numerous edge servers deployed at the network edge, and a huge number of distributed end devices. Instead of considering them as separated parts, most applications require all of them to be well orchestrated for providing reliable services over different temporal and spatial scales. For example, in airport surveillance applications, the edge servers can analyse and filter the video streams before uploading the whole batch to the central server, which can significantly decrease the traffic flow over the WAN and relieve the burden on the central server without performance loss. Furthermore, considering a scenario where an end-user requests some content of interest from an edge server offering caching services, the central server can work as a complementary to the edge servers with limited storage capacity.

Several works survey cloud computing or edge computing from different perspectives. In Reference [68], the authors discuss the architecture and performance optimization approaches in mobile cloud computing, which employs centralized cloud servers to offer computation offloading and storage for mobile users through WAN. Reference [174] surveys the computing techniques for big data analytics, including cloud computing, TC, and fog computing. It is believed that analyzing data at the network edge, rather than the central cloud server, may be a better solution in an IoT context, due to the constrained bandwidth of WAN and the context-aware requirement. In

Reference [113], the authors discuss the existing works on computation task offloading in an edge computing paradigm, mainly focusing on the tradeoff between energy consumption and delay. Reference [150] surveys the works of caching strategies in the context of radio access networks (RANs) enhanced by edge computing.

Although the above-mentioned surveys are inspiring, none of them pay specific attention to research issues in the hierarchical computing architecture, which provides great benefits offered by the orchestration of end devices, the edge, and the cloud. The scope of this survey covers a comparison of different network computing paradigms and different research issues, including computation offloading, caching, security, and privacy under the hierarchical computing architecture. The remainder of this survey is organized as follows. Section 2 introduces and compares the emerging computing paradigms. Section 3 reviews the research of computation offloading in the emerging computing paradigm. Caching strategies and security and privacy protection mechanisms are summarized in Sections 4 and 5, respectively. Section 6 outlines some potential future directions in this emerging filed of study, followed by a conclusion, given in Section 7.

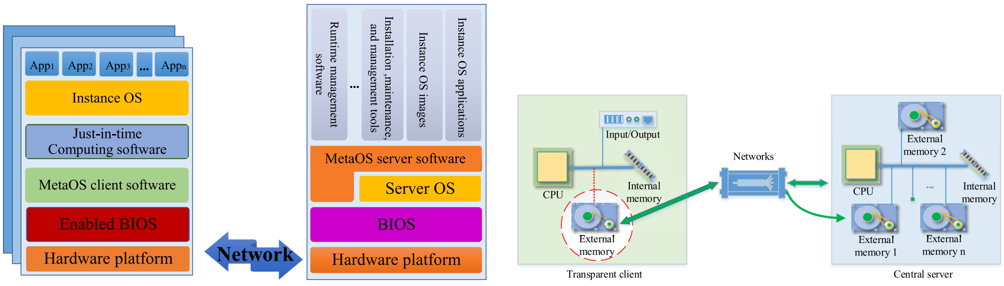
2 EMERGING COMPUTING PARADIGMS AND EVALUATION CRITERIONS

In the past few years, to address massive data computation, various computing paradigms have been proposed to provide timely and resource-efficient services. In this section, we introduce the computing paradigms emerging in recent years, including transparent computing (referred to as TC, hereafter), fog computing (referred to as Fog), mobile edge computing (MEC), and cloudlet. Although they face some common issues to deal with the huge amount of computing and storage tasks generated by heterogeneous devices, such as the management of computing and storage resources and networking, they also exhibit different characteristics due to various original driving forces, such as persuasive computing for TC and IoT applications for Fog computing. For instance, all these computing paradigms need to mask the heterogeneity of various devices to ease the resource management. To this end, TC adopts the operating system–(OS) level solution, e.g., Meta OS [178], while the other three paradigms focus on virtualization and containerization solutions.

2.1 Transparent Computing

Transparent computing was proposed to decouple the software, including operating systems (OSes), from the heterogeneous hardware of IoT devices [172]. It masks the details of service provisioning and serves the users in a totally "transparent" way [123]. To this end, TC enables devices to choose services on demand via networks, without considering the details of service provisioning, such as the upgrade and management of softwares. Generally speaking, TC can be characterized by the following properties:

(1) Working in a client-server mode, TC logically integrates the devices distributed across the network as one system. The system intelligently provides services according to the capabilities of devices and the conditions of the networks. The server side is in charge of the centralized resource management for the network-connected clients to provide elastic services.
(2) To logically split the software from the hardware of heterogeneous client devices, TC develops an on-demand service loading and execution architecture, empowering the client devices with the capability of dynamically executing cross-platform services from remote servers via high-speed networks.
(3) To fully use the computation resources residing in the client devices, TC enables client devices to fetch remote services from the server side on-demand and execute them locally in a block-streaming way. Block-streaming execution means that when users request a



(a) The Layers of a TC system

(b) The extension of von Neumann architecture

Fig. 1. An illustration of the TC architecture.

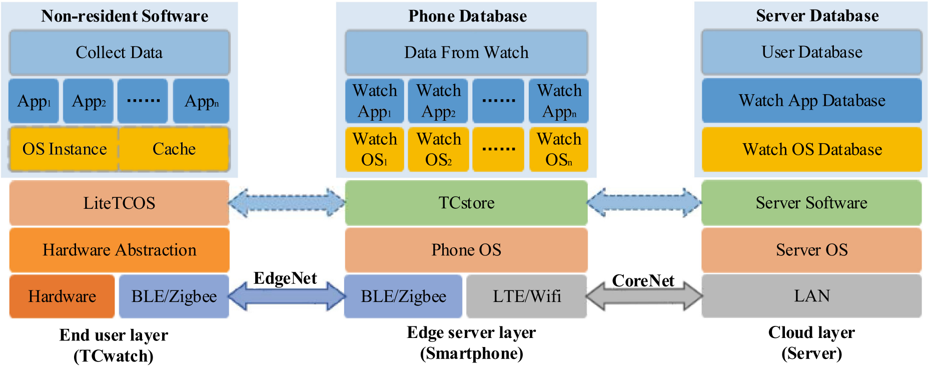


Fig. 2. An illustration of TC-based IoT system.

specific service, only the necessary part of codes related to the service, rather than the whole software, will be loaded and executed on client devices. Consequently, the energy efficiency and delay of service provisioning can be significantly improved.

We describe a typical implementation of TC in Figure 1. It extends the classical von Neumann architecture in both spatial and temporal domains. In the architecture of TC, a single computer is extended to network-connected computers/devices. As shown in Figure 1, the servers store the system/service software that can be dynamically loaded to client devices for execution via the Meta OS platform. Built on an underlying OS, the Meta OS is designed to shield the heterogeneity of hardwares and unifies the interfaces to upper layers. In such a way, various commodity OSes and applications can be initialized and managed by the Meta OS platform. The Just-In-Time Computing layer is designed to enable the client devices to load the instructions of the demanded programs and user data from servers through block-streaming. After remote loading, the client devices can perform the computation with the local resources of client devices in a timely manner. Under this architecture, the storage of the client device is extended to other network-connected devices (or servers), and the I/O interrupts are redirected from the system bus of the local device to the network [173].

Recently, the advantages of TC have been further recognized in the IoT era, which urges a well-designed solution to manage the huge amount of software for highly heterogeneous hardware infrastructures, ranging from high-end servers, laptops, and smart phones to low-end sensors. Ren et al. [123] propose a scalable TC-based IoT architecture, as shown in Figure 2, to provision flexible

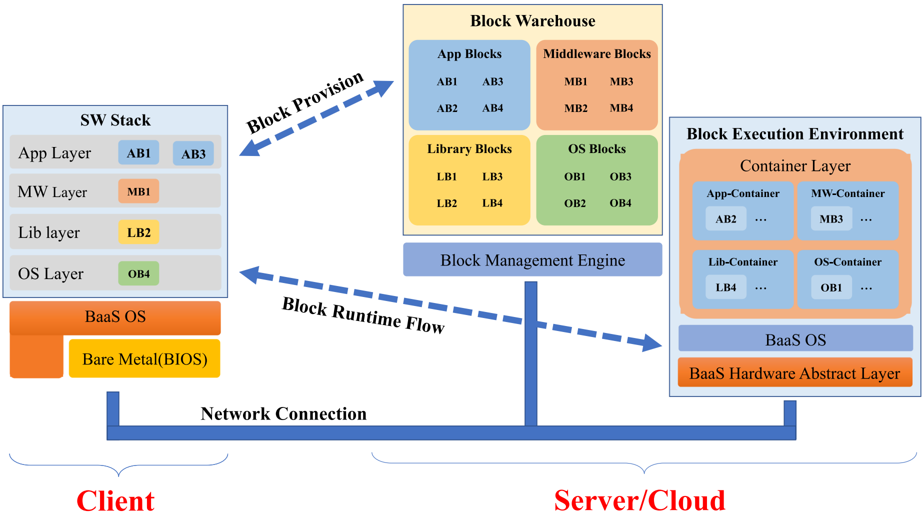


Fig. 3. A typical BaaS architecture.

and timely services at the edge of the network. In their proposed architecture, nearby edge servers act as the TC server to store some frequently used Apps and data and achieve timely response for the service requests of IoT devices, while the cloud acts as the storage and management center to provide centralized control for the whole system. A LiteTCOS is developed as the MetaOS for IoT devices and is responsible for commodity OS loading and block-streaming service execution. Moreover, He et al. [67] propose a new service model based on TC, named Block-stream, as a Service (BaaS) to achieve ambient service computing for IoT devices. They present a clear BaaS architecture, as shown in Figure 3, where any kind of software, including OS, libraries, middleware, and Apps, is divided into a set of code blocks. When the server receives a service request, part of the code blocks will be dynamically provisioned to IoT devices for timely and efficient response.

2.2 Mobile Edge Computing

MEC is initiated by the European Telecommunication Standards Institute to enable cloud computing services in proximity of the mobile subscribers [9, 27, 96]. By deploying MEC servers at the macro or micro base stations, MEC can improve the user experience by processing the user request at the network edge with reduced latency and location-awareness, as well as alleviate the load over the core network, as shown in Figure 4 [12]. Together with network function visualization and software-defined network, MEC has been deemed as a key enabling technology toward the 5G era [70, 167].

To stimulate the seamless involvements of vendors, service providers, and third-party players on MEC, an industry standardization group has been established in ETSI to develop specifications for a standardized and open MEC environment. The members of ETSI MEC ISG include Huawei, Intel, Nokia, Vodafone, NTT DOCOMO, and so on. The first introductory technical white paper was published in 2014 to specify the concept of MEC and the reference architecture of MEC platform [1]. Furthermore, it also discusses the key enabling techniques and challenges in MEC. During 2015 to early 2017, the group has documented several specifications of MEC, ranging from the terminology, service scenarios to technical requirements in MECs [2, 4, 5, 9]. Although the group claims that MEC mainly focuses on the integration of cloud computing technology into cellular

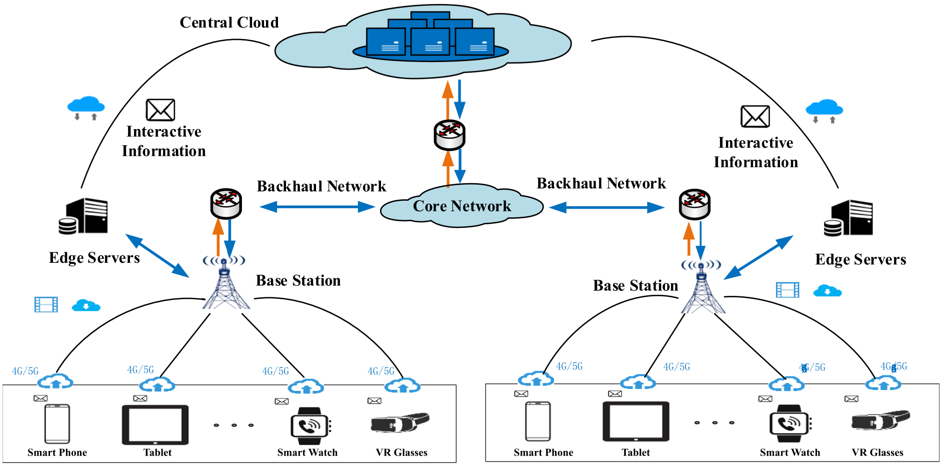


Fig. 4. Architecture of mobile edge computing.

networks in the early age, the name of MEC is now changed to multi-access edge computing, which can reflect the growing interests from non-cellular operators [3].

In recent years, numerous solutions have been proposed by both academia and industry to enhance the performance of MEC, such as modeling, multiuser resource allocation, and system implementation, and so on. Several surveys have focused on the progress of MEC from different perspectives [12]. Some real-time MEC application scenarios are discussed in References [12, 96, 113, 139, 150]. They also discuss the taxonomy of MEC from different viewpoints, such as the characteristics, actors, access technologies, applications, key enablers, and so on. A survey on the fundamental key enabling technologies of MEC is presented in Reference [139]. It discusses MEC orchestration by taking both system performance and MEC platforms into consideration, shedding light on the different orchestration deployments. In addition, the authors also introduce the architectures and typical deployment scenarios of MEC. The authors of Reference [96] survey the state-of-the-art MEC studies, focusing on the joint management of communication and computation resources. The survey in Reference [113] discusses the key use case in MEC, i.e., computation offloading. The authors of Reference [150] survey the key enablers of MEC, including cloud computing, SDN/NFV, and smart devices. Besides, they also discuss some key technologies in mobile edge networks, covering cloud technology, and SDN/NFV, as well as smart devices.

2.3 Fog Computing

The Fog computing paradigm was first coined by Cisco in 2012 [22]. In some sense, fog computing is similar with the concept of MEC. However, it is also a novel network computing architecture that provides the capabilities of computing at the network edge [27]. Fog computing is originally proposed for the context of IoTs, which demands location awareness and timely response in addition to wireless access and mobility support. In addition, fog computing uses an n-tier architecture to offer more flexible services, which highlights that all the network devices along the data routing path can provide data computing and storage services for end devices. As shown in Figure 5, a fog tier is physically placed between the cloud and the IoT devices to enable compute, storage, and networking resources pooling. The fog tier consists a large number of heterogeneous nano servers, ranging from dedicated devices such as edge routers, set-up boxes, and temporary devices such as smart phones, high-end sensors, and vehicles [21].

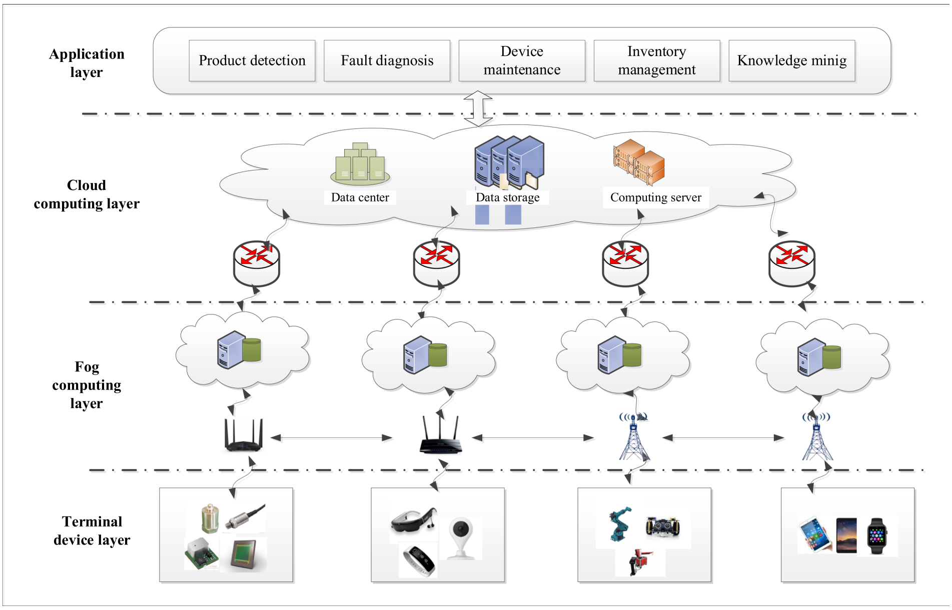


Fig. 5. Architecture of fog computing.

Typical application scenarios of fog computing are the ones that require both real-time control to improve the instantaneous system performance and long-term batch data analytics to gain insights for business strategy adjustment, which calls for the interplay between fog servers and the cloud server. Real-time control demands the densely distributed fog servers to reduce service latency, and the batch data analytics naturally falls into the expertise of cloud servers with massive computing and storage resources. To promote the development of fog computing, the Openfog Consortium was founded in 2015 by the leading companies in the IT industry, including Cisco, Intel, Microsoft, Princeton University, Dell, ARM, and so on. Numerous use cases of fog computing can be found in Reference [21], ranging from smart traffic lightweight system to wind farms and smart grid.

Much of the literature has surveyed the research issues of fog computing with different points of focus [27, 144]. Vaquero and Rodero-Merino [144] overview the enabling technologies and future development of fog computing. The survey in Reference [161] overviews the fog computing definition and some typical application scenarios, as well as clearly presents the challenges in fog computing system design and implementation. The authors of Reference [160] claim that fog computing is expected to be a natural platform for many promising and challenging IoT scenarios. The authors of Reference [47] describe the advantages of fog computing for IoT and introduce various application scenarios that are suitable for fog computing. The authors of Reference [104] put forward 10 questions and give out the corresponding answer to demonstrate the advantages of fog computing compared with the other existing computing paradigms. More recently, the authors of Reference [27] propose a concise set of evaluation criteria in fog computing.

2.4 Cloudlet

The concept of cloudlet was first proposed in 2009 by a research team from Carnegie Mellon University [130]. The term “cloudlet” refers to micro data centers that are placed in the proximity of mobile users, e.g., in a coffee shop or a classroom. The key motivation behind a cloudlet is to boost the interactive performance of mobile applications, especially the ones with stringent

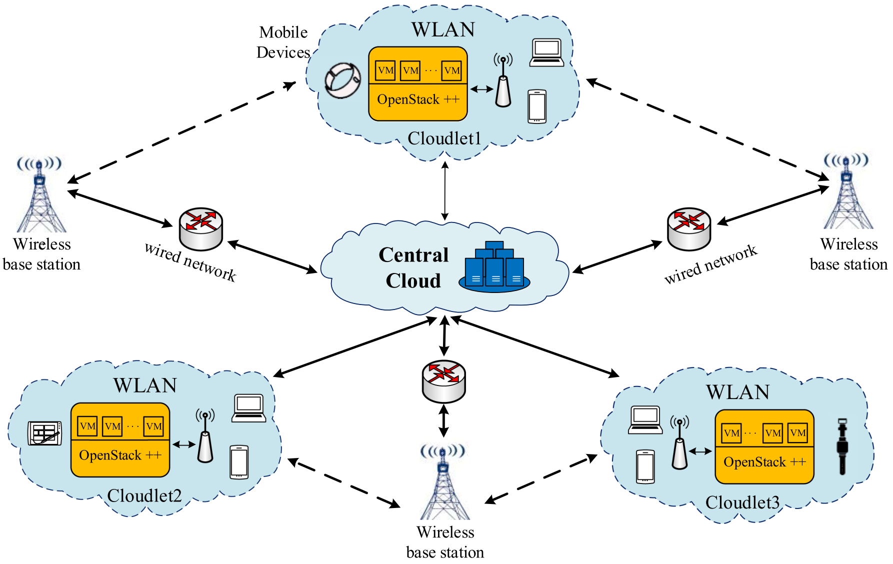


Fig. 6. Architecture of cloudlet-based computing [131].

requirements on end-to-end latency and jitter. Such applications demand that the responding delay is in the order of milliseconds, which is unlikely to happen through the Internet. To fill this gap, the proximity of cloudlets enables the servers to provide highly responsive cloud services to mobile users and hence complement the three-tier cloud hierarchy, i.e., mobile users-cloudlet-cloud, as shown in Figure 6. In addition to provide timely service, this hierarchy enforces the privacy of its owners by denaturing the private data before releasing them to the cloud [131]. Several typical use cases of cloudlets have been identified, such as assisted driving support, sports training assistants, and so on.

After the invention of the cloudlet, the research group in CMU has published a series of works mainly focusing on two subjects, i.e., identifying the valuable applications for cloudlet [129] and designing virtual machines to support user mobility [66]. In Reference [129], the authors state that real-time cognitive assistance, e.g., face and speech recognition, can be the "killer app" for the cloudlet, which demands timely response to unobtrusively guide users' attention. Based on an open source ecosystem for cloud computing called OpenStack, Reference [66] proposes the design of a VM overlay to enable cloudlet discovery and hand-off in the mobile context, while minimizing data exchange among cloudlets. Due to the potential business opportunities brought by cloudlets, CMU and several leading industrial companies, including Intel, Nokia, Crown Castle, Vodafone, and Deutsche Telekom, formed the open edge computing initiative [108], aiming at developing key technologies surrounding cloudlets and conducting user acceptance testing.

Instead of deploying dedicated computing facilities in the proximity of users, several existing works utilize the idle computing resources of mobile devices to perform computing services, which is termed "Ad hoc cloudlet" [36]. In an ad hoc cloudlet, the mobile devices connect with each other through short-range radio communication technologies to perform data analytics for IoT devices with limited computing capabilities, such as sensors for RFIDs. Although utilizing the idle resources of mobile devices requires minimum efforts on facility deployment and maintenance, the mobility and dynamic local load of mobile devices impose significant challenges to enable efficient computing and service provisioning in an ad hoc cloudlet. To address this issue, some

Table 1. Similarities and Differences Comparison of Various Computing Paradigms

Paradigms	Location for Computing	Virtualization	Research Focuses
TC	End devices and nearby devices	MetaOS	Cross-platform and on-demand service provisioning, and so on
MEC	Base stations and nearby devices	VM and container	Uplink computation offloading downlink caching, and so on
Fog	Devices along the routing path	VM and container	Uplink computation offloading, downlink caching, and so on
Cloudlet	Nearby cloudlets	VM	Computation offloading, VM management, and so on
Cloud	Central Cloud	VM	Workflow scheduling, VM management, and so on

works design stochastic optimization approaches to improve the adaptability of ad hoc cloudlets to system dynamics [119].

2.5 Discussion of the Similarities and Differences

The common features of the aforementioned emerging computing paradigms are to reduce the complexity and cost of hardware implementation, to reduce the latency, as well as to improve the quality-of-experience and network efficiency. However, there are some subtle differences that need to be clarified, as illustrated in Table 1.

First, the motivation of the TC paradigm is to address the heterogeneities of various end-users in IoT networks and support different OS and cross-platform application software. Thus, the complexity and cost of hardware implementation are reduced and the compatibility of applications are enhanced. On the contrary, the motivation of the introduction of fog, MEC, and cloudlet paradigms is to process data at the proximity of end devices to reduce the response delay and improve user experience and network efficiency. Note that cloudlets mainly use virtual machine for virtualization, while MEC and fog consider containers.

Second, for the TC paradigm that emphasizes providing service-oriented computing solutions, the computation is executed at the end-users/edge-nodes via acquiring the OS and application software from the edge/cloud server. Moreover, by block-streaming service execution method, lightweight terminals can only execute the necessary codes rather than the whole program of an application to enhance the energy efficiency of service computing. For Fog, MEC, and cloudlet paradigms, in general, end devices and edge servers all install OS and specified application software. Edge servers can provide timely data processing to the service requests of end devices, while end devices can improve their energy efficiency by offloading some computation-intensives tasks to the edge servers or the cloud server.

Third, from the perspective of security, these computing paradigms have different research focuses. TC aims at leveraging the computing capabilities of end devices and edge servers to provide secure services for IoT devices. Security mechanisms are generally deployed at the MetaOS layer (both of the end device side and the edge server side) to detect malware and OS-level attacks, because MetaOS works under the commodity OS and has the higher priority to directly check the program codes running in the hardware by occasionally hanging up the upper-layer OS. But for fog, MEC, and cloudlet paradigms, researchers mainly focus on securing the computing environment of edge servers and offloading the security burden (such as encryption and authentication) from end devices to edge servers. Therefore, finding ways to leverage the orchestration of end devices, edge servers, and the cloud to design sophisticated security solutions becomes a hot research trend for all the network computing paradigms [124].

2.6 Evaluation Criteria

The full benefits of the aforementioned emerging computing paradigms are leveraged on some common performance criteria for investigating the architecture and the optimization algorithms of these networks [27].

The first criterion (C1) is the need to support heterogeneity in resources. In emerging computing paradigms, different access nodes at the edge of network and end-users exhibit strong heterogeneity in terms of computational and storage capabilities. Therefore, one needs to take the heterogeneity of various access nodes into account for designing the computing architecture and optimization algorithms. For example, the edge server of transparent computing needs to provide various OS and application software to support a variety of end-users, which generally have different hardware configurations. To accommodate heterogeneous services over heterogeneous end-users, Intel provides an HTML5-based solution for achieving TC. Compared with the previous versions, HTML 5 possesses some new features, such as rich semantic information and multi-thread support. Although HTML5 possesses the advantages of cross-platform support and low development costs, challenges remain due to the differences between browsers and Web security. The TC paradigm enables the implementation of browser engines at the Meta OS layer to shield the differences between browsers [173].

The second criterion (C2) is the stringent QoS requirements in the emerging computing paradigms. In future communication network, QoS requirements, such as low latency, high data transmission rate, high spectrum, and energy efficiency, are the key performance indicators for evaluating the performance of new communication and computing technologies. The key motivation of introducing emerging computing paradigms is to reduce the data delivery latency while increasing the data delivery rate [49]. For example, the delivery latency needs to be smaller than 10 ms in vehicle-to-vehicle communication or intelligent vehicular communication network, and the transmission rate needs to be larger than Gigabits for virtual reality. Therefore, one needs to take these requirements into consideration for the design of the network architecture and the optimization algorithms of emerging computing paradigms.

The third criterion (C3) is the need for elastic scalability. The main goal of the emerging computing paradigms is to realize low latency, high data rate, and high energy efficiency for a large variety of end-users in IoT scenarios. The computing paradigms are expected to provide services for millions or, even more, of end-users and applications. Accordingly, they need to have the ability to provide an elastic on-demand service for a variety of terminals. It also means that the architecture and the corresponding algorithms of the emerging computing paradigms need to have the capabilities of adapting to the changes in network scale.

The fourth criterion (C4) is whether the computing paradigm can support mobility. For example, in vehicular networks, the movements of vehicles significantly affect the performance of the communication system. In particular, the fast changes of the channel need to be carefully considered for emerging computing applications.

The fifth criterion (C5) is the need of federation and interoperability. In the emerging computing paradigms, the edge servers have geographically distributed deployment on a wide scale and are provided by different owners. The cooperation among different providers can facilitate the usage of various services and improve the experience of end-users. To this end, it calls for well-designed data and control interfaces that can enable interoperability at different levels of providers and architecture. For example, in a TC paradigm, Meta OS is responsible to manage hardware and software resources, including OS and various application software (Apps), and to provide computing services for users securely and reliably [172].

3 COMPUTATION OFFLOADING FOR EMERGING COMPUTING PARADIGMS

To liberate mobile devices from the limited computation capabilities and energy supply, both industry and academia take computation offloading as a promising solution to efficiently harmonize the computing resources in the end-edge-cloud orchestrated computing paradigms [168]. The resource-limited end devices, such as mobile phones, can offload the compute-intensive tasks to the powerful facilities to either the edge servers or remote cloud servers to converse energy consumption and reduce the response time. In the following, we divide the existing literature into three categorizes according to their objectives, i.e., minimizing energy consumption, minimizing delay, and jointly optimizing energy consumption and delay.

3.1 Minimizing Energy Consumption

Research reports have shown that information and communication technology (ICT) sector is responsible for 0.75 million tons of CO_2 gas emissions for 1 TWh of energy consumption. Motivated by this observation, energy-efficient computation offloading has attracted extensive attention. Considering the energy consumption caused by both code compilation and execution, Chen et al. design an offloading scheme to strike the balance between computation and communication in Reference [33]. A partial offloading scheme is proposed to conserve energy of each MD for context interactive applications, taking into account the social relationships between users [25]. In Reference [46], the authors present an energy-aware offloading approach to enable fine-grained code offloading while bringing minimal burden on the programmers. This work determines the decision of computation offloading at runtime. The above surveyed works focus on offloading to a single cloud server. In Reference [107], the authors propose an energy-efficient multisite offloading approach, in which the partitioned applications can be executed by either the MDs or several servers. The application is modeled as a weight object relation graph, in which a edge weight represents the communication energy cost, and a node weight indicates computation energy cost. The objective is to optimize the energy consumption. In Reference [24], the authors provide a strategy to reduce the overall energy consumption without sacrificing the network performance. In Reference [153], the authors involve a small cell cloud manager into the edge computing architecture that manages the computing-related activities of the femto-cloud. The authors jointly optimize the allocation of communication and computation resources based on partial offloading. Moreover, dynamic voltage scaling technology is used to adjust the computational speed of MDs to reduce energy cost or reduce execution time.

Numerous works take the dynamics in wireless connections between MDs and servers, and the available computation resources in servers into consideration. In Reference [54], the authors propose an offloading decision framework to choose an optimal resource provider, such as a local MD, a cloudlet, or the remote cloud. The framework consists of a profiler, context manager, offloading decision maker, execution planner, and distributed service execution engine. The profiler analyses the characteristics of the operations and resource consumption profiles, based on the context information gathered by the context manager. The execution planner studies possible computation policies based on data locations and context information, while the decision maker chooses the optimal scheduling policy and resource provider. Reference [35] finds the optimal service mode by the cooperation among cloudlets and remote cloud. It takes the mobility of cloudlets into account in the cooperative service provisioning. For example, co-located clouds-based service can provide fine-grained mobility support at the cost of potential lower computing capability in comparison to the powerful remote cloud. The authors schedule the computation offloading in an opportunistic way to enable high mobility while minimizing cost. Different from traditional service modes of remote cloud and cloudlets, the proposed method can achieve a flexible tradeoff between energy cost and mobility support. The optimal option of task schedule is to minimize energy cost. Differently from Reference [35], Reference [54] organizes MDs into clusters to provide cloudlike services.

In Reference [26], the authors consider mobile computing offloading for heterogeneous networks. This article supposes that there are multiple offloadable components in one application, each of which has different size and computation complexity. To conserve the energy of the MD under a given delay constraint, it presents a combinatorial optimization algorithm to make offloading decisions. The delay includes both communication delay and execution delay. The optimal algorithm can get around 43% energy savings. In Reference [48], the authors consider the scenario where small cells that equipped with computing and storage resources connect MDs through a high-rate wireless channel. Mobile applications can be partitioned into several components and some of these components can be offloaded to the small cells. Differently from Reference [26], when making the offloading decision, the authors take the dependency relationship among components into consideration and formulate it as a generic graph. Then, they propose to minimize the energy cost of MDs with a strict delay requirement, including communication delay and execution delay. In Reference [169], the authors consider 5G heterogeneous networks, including a set of MDs and a macro base station built-in with an edge server. They present a multi-user computation offloading framework to minimize the system energy cost with the delay constraints by jointly optimizing the offloading decision and the radio resource. Reference [65] investigates the computation offloading in a three-level hierarchy, in which the MDs are connected to edge servers through wireless links, and the edge servers connect with the cloud server by optical fibers. To minimize the energy consumption of MDs while maintaining the delay under a threshold, the authors propose both a centralized optimization scheme and a distributed offloading scheme based on game theory.

In addition, adaptive power control and renewable energy exploitation are powerful methods to conserve energy consumption. In Reference [93], the authors propose a distributed power control algorithm by fine-tuning transmission power for the small cell base stations. Consequently, it efficiently improves the delivery ratio to end users within required delay. In Reference [97], the authors consider a MEC system consisting of a MD equipped with an energy harvesting component and a MEC server. The energy cost minimization problem, which incorporates both the execution delay and offloading failure, is formulated as a high-dimensional Markov decision problem. An online Lyapunov optimization algorithm is proposed to jointly decides the CPU frequencies, the transmit power, and the offloading decision for computation offloading. The simulation results demonstrate that the proposed algorithm can significantly improve the performance in terms of energy consumption and effectively decreases offloading failure. In Reference [40], the authors consider a three-layer computing architecture, consisting of one MD, one edge server at the edge, and one remote cloud server. The edge server decides whether to process the tasks or further offload it to remote clouds, with the objective to minimize the weighted sum of energy consumption and delay. To address this problem, an efficient heuristic algorithm are proposed by using semidefinite relaxation and a randomization mapping approach. Simulation results illustrate that the proposed approach enables substantial improvements via using a computing access point between the remote cloud and the MD.

References [97, 165] consider the cases where the MDs can be powered by energy harvested from the ambient energy sources to further prolong their lifetime. In Reference [97], Mao et al. schedule an energy harvesting–(EH) powered MD to offload its task to a proximate edge server. Both the CPU speed and transmission power are optimized to minimize the long-term cost that combines the execution delay and the penalty caused by task dropping. In addition to EH-powered MDs, Zhang et al. [165] consider that the execution of a task may exceed one time slot, which leads to the coupling of task processing across slots. To address this issue, Reference [165] divides the original tasks into smaller subtasks that can be processed in each time slot.

3.2 Minimizing Delay

One of the main targets of end-edge-cloud orchestrated computing paradigms is to reduce the service delay. In Reference [44], the authors present an approach to optimize the total execution time of an application composed of multiple modules. This work introduces a dynamic application partitioning mechanism between the user equipment and the cloud. In Reference [59], the authors model the execution of applications as graphs and determine the optimal distribution of the application between servers and MDs. First, they abstract the behavior of application modules as a dataflow graph. Each module offers a set of services and connects with each other. By offline profiling, the dependencies among modules are characterized by their resource cost and hence can provide some *a priori* knowledge for optimization. Second, a partitioning algorithm is performed to minimize the interaction delay. In Reference [159], the authors investigate multi-user application partitioning problem and further consider how to partition jointly computations of multiple users. Instead of minimizing the service completion time for each user, the work aims to minimize mean completion time for all users.

In Reference [86], the authors consider the computation offloading problem in an MEC system comprising an MD and an MEC server. The authors find that the execution time of most mobile applications is in the range of tens of milliseconds, while the typical duration value of a channel block is a few milliseconds. It implies that the application execution process may be across many channel blocks, transforming the computation offloading problem as a two-timescale stochastic optimization problem. Specifically, the offloading decision is made in the large timescale, and the transmission policy is made in small timescale by taking the instantaneous wireless channel conditions into consideration. Based on such insights, the work analyses the average delay of each task and proposes an efficient algorithm to minimize the offloading delay. To minimize the mean delay of general traffic flows in the LTE downlink, in Reference [57], the authors introduce a mobile edge scheduler for MEC paradigm. It is deployed closely to the eNodeB and implemented with a channel-aware and flow-aware scheduling policy. By accommodating the transmissions to the available channel quality of MDs, it can minimize the mean delay for the complete set of traffic flows.

Due to several stochastic factors, such as the changing wireless connections, fluctuation of transmission bandwidth, and user mobility, data transmission between the mobile user and the cloud is highly unreliable. In Reference [134], the authors demonstrate that the "bad" connectivity consumes a lot of energy at MDs. To cope with this problem, they propose a Lyapunov optimization to optimize response delay, only requiring prior system knowledge. It adaptively utilizes the duration of good connectivity to prefetch frequently used data while deferring delay-insensitive data in bad communication status. Generally, users have different service delay and energy consumption requirements for various applications. In Reference [152], the authors design an optimization method to minimize the weighted sum of energy consumption and the computation delay. The proposed method consists of two steps. First, they determine whether it should be offloaded or not. Second, it should be offloaded to a particular server. Reference [88] leverages the social relationships between MDs to design a Nash equilibrium-based solution. The objective of Reference [88] is to minimize the social group execution cost, which is defined as the sum of the execution delay and the punishment caused by task dropping. Reference [82] designs a deep reinforcement learning–(DRL) based algorithm to deal with the instability in a device-to-device offloading scenario. The DRL-based algorithm takes the length of task queues at MDs and edge servers, and their distances as the input, and outputs the computation offloading decisions to the user utility obtained by task execution, while maintaining the energy consumption and delay under given thresholds.

The aforementioned works focus on how to offload the application workloads from MDs to edge clouds. However, how to select an optimal edge site in the edge network to undertake the workloads is also critical to optimize the response delay of mobile applications. For example, users may conglomerate close to a single cloudlet but far away from another. However, this situation makes an overloading usage of the first cloudlets resources, while resulting in a wasted capacity of the second. For this scenario, in Reference [127], the authors propose an approach for minimizing offloading delay with two cloudlet servers. The article improves not only the computation delay by VM migration but also the communication delay by transmission power control. In Reference [138], the authors further explore this problem and considered the scenario that MDs offload their application to geographically distributed cloudlets. To achieve a minimization of the average response time, they propose a delay-aware task offloading strategy to allocate MDs' computation tasks into optimal cloudlets. The cost is both considered the network delay and the cloudlet delay.

The remote cloud can cooperate with the edge to achieve better delay performance. In Reference [29], the authors consider three types of actors consisting of MDs, a cloud edge, and a data center cloud, to comprise a resource pool. This article exploits the Lyapunov optimization approach to manage resource pools for improving the overall experienced delay of mobile users. Similarly to Reference [29], in Reference [179], the authors consider a scenario with multi-users, one resource constrained local cloud and one resource-abundant remote cloud. According to the requirements of applications in terms of delay, the authors design a priority queue-based threshold policy to maximize the probability of completing application in time. Numerical results demonstrate that the quality of service is vastly improved with the cooperation of remote cloud when the task queue of edge cloud is exceeded. In Reference [89], cloudlets can determine whether to respond user service requests locally or offload them to the remote cloud. The authors propose a multi-resource allocation strategy to improve the QoS. Moreover, to maximize the long-term reward under service delay requirements, they formulate a semi-Markov decision process problem and address it by linear programming.

Usually, increasing workload leads to an increasing delay. To reduce the delay, the user may eventually increase the transmission rate at the expensive of increasing radiated power. Accordingly, the system workload needs to make an endeavour on the tradeoff between the delay and the energy consumption. In Reference [105], the authors consider a system with multiple users served by a small cell node with computational capabilities, and partial instructions of an application can be offloaded to the small cell node. The authors investigate the delay and energy consumption tradeoff through adjusting uplink data transmission rates. Their simulation results demonstrate that, for a given energy cost, the delay increases with increasing number of users. As a result, the uplink rate should be increased to achieve certain quality-of-experience by sacrificing energy efficiency. Reference [98] considers the joint optimization of delay and energy consumption during the computation process in a multiuser scenario. An online algorithm is designed to decide whether local execution or computation offloading, with respect to the constraints on task buffer stability. The proposed algorithm determines the optimal CPU frequencies of MDs in each time slot and the optimal transmission power and bandwidth using the Gauss-Seidel method.

3.3 Joint Optimization of Energy Consumption and Delay

To improve the performance of service provisioning in the end-edge-cloud orchestrated computing paradigms, some literature investigates computation offloading by taking both the energy consumption and delay into consideration. In Reference [84], the authors propose a task scheduling algorithm that jointly schedules the cloud resources and wireless channels to minimize the energy consumption with constraints on completion time. Lin et al. describe an application as a directed acyclic graph, in which a node and an edge represent a task and the execution sequence, respectively. In Reference [171], the authors partition an application into a sequence of tasks, aiming to optimize the energy consumption of the MD under a execution deadline constraint. They further study collaborative execution between the MD and the cloud by formulating the execution process as a constrained shortest path problem. In Reference [94], the authors minimize the energy consumption of MDs under the constraints of execution delay and component precedence ordering. They design a wireless connection-aware offloading algorithm for multi-component applications to decide the offloaded components of the application.

In Reference [180], the authors consider the scenario consisting of multiple femtoclouds with computation and storage capabilities and MDs, each of which has one task to execute. The application can be partitioned into two parts, in which one part is executed locally and the other is executed at the edge. Targeting at minimizing the overall energy cost of MDs under delay constraints, this work jointly optimizes the allocation of radio and computational resources. Typically, the proposed approach can reduce 40% average total energy when compared to the no-offloading solution, and the time complexity of the proposed approach is only $O(K)$. In References [162, 163], the authors also consider a multiuser system that consists multiple single-antenna MDs and a single-antenna base station. The computation offloading problem is formulated as a convex optimization problem to minimize the weighted sum of multiusers' energy consumption under delay constraints.

In Reference [85], the authors define a customizable cost model, which enables users to adjust the weight of delay and energy in the optimization problem. In Reference [43], the authors partition and offload applications at runtime to optimize the delay and energy consumption. Similarly to Reference [43], Reference [80] proposes a framework called ThinkAir, which utilize the smartphone virtualization technique to facilitate the application mitigation from smart phone to cloud. Reference [20] considers the case where the MDs are powered by wireless energy transferred from the edge servers; the authors schedule the time allocation between energy transfer and computation task offloading to maximize the computation rate, i.e., the ratio between processed data and the processing time. To account for the the selfishness of MDs, Reference [76] designs a game theory–based scheme to jointly optimize the energy consumption and delay of computation offloading. Furthermore, the authors analyze the price of anarchy of the scheme, which quantifies the gap between the proposed scheme and the optimal solution.

Pu et al. [119] investigate energy-efficient computation task offloading among multiple MDs through cellular and D2D links. The authors consider the stochastic task arrivals and channel conditions over time and propose an online algorithm to dynamically offload the tasks to minimize the long-term energy consumption. References [34, 99, 155] design online algorithms to strike the balance between energy consumption and execution delay. Considering an MEC system consisting of multiple users and edge servers, Mao et al. [99] minimize the long-term average weighted sum power consumption of the MEC system by adjusting the radio and computation resources at both the MDs and edge servers. Chen et al. [34] investigates the computation offloading among the edge servers deployed at small-cell base stations. The problem formulation takes the long-term delay cost and energy consumption of edge servers as the objective function and constraints, respectively. Wu et al. [155] investigate the computation offloading in a three-level hierarchy consisting of MDs, edge server, and a remote cloud server. Considering the different bandwidth and computation resources among the three levels, the authors propose an online algorithm to minimize the average energy consumption of MDs while ensuring the response time under a given time constraint.

In multiuser scenarios, one critical factor that affects the performance of computation offloading is the wireless access efficiency. If too many users select the same wireless channel to connect the edge access node simultaneously, then they may suffer severe interference. Reference [41] investigates the computation offloading in an interference environment. First, the authors model the offloading overhead with a tradeoff factor, which denotes the weighting parameter of computational time and energy. Then, they formulate the multiuser offloading decision problem as a game and propose to achieve the Nash equilibrium. Based on Reference [40], the authors of Reference [39] further extend the study to multi-user scenario. Different from Reference [40], this work aims to jointly optimize communication and processing resource allocation among competing MDs. However, References [40, 179] only consider the offloading policy from MDs to one server. In Reference [63], the authors consider a heterogeneous multi-site offloading environment, consisting of MDs, cloudlets, and public clouds. The cloudlets act as distributed proximal devices, such as WiFi access points, Femtocell access points, and Macro-cell access points. These devices might be heterogeneous in terms of processing speed, communication delay, disconnection probability, and so on. Public clouds are assumed to have similar transmission bandwidth and processing speed. Tasks are offloaded to these heterogeneous clouds or cloudlets. The authors propose a multi-site computation offloading algorithm to achieve an optimal offloading site for each task to minimize the overall cost of energy consumption and execution time. It is noteworthy that fog radio access network is the evolved network by equipping the RRHs with caching and signal processing facilities to improve the spectral efficiency and reduce the delivery delay between the baseband units and the RRHs. The work in Reference [111] optimizes the delivery phase involving both fronthaul communications and wireless transmissions by considering channel precoding and cached content.

4 CACHING FOR EMERGING COMPUTING PARADIGMS

In the emerging end-edge-cloud orchestrated architecture, nodes at different levels of network have different abilities to compute and store data files. At the same time, different caching strategies have different impacts on the system performance, such as the network throughput, system energy efficiency, delivery latency, and so on. However, the dynamic characteristics of the user behaviours, user demands, and network environments make the caching decision very challenging. Consequently, the research on the cache placement, contents, and strategies has received numerous attention from both academia and industry for the end-edge-cloud orchestrated architecture.

4.1 Placing Caching Units

In the end-edge-cloud orchestrated architecture, edge devices should be able to provide various application services and cache the relevant data for computation-intensive end-users. To give full play to the abilities of edge devices, the cache placement should be related to the capabilities and locations of edge devices as well as the target severed [17, 30, 38, 73].

For a transparent computing paradigm, the distributed clients need to load the image of OS and software stored on the centralized transparent severs to perform the computing task via the network. A great deal of requests from distributed clients for fetching OS and application software results in the bottleneck of access to servers. Therefore, the transmission delay between servers and clients in transparent computing should be carefully considered to improve the system performance. To address the problem, Reference [58] designs a two-level cache structure composing of block caches of servers and clients. The authors exploit client cache to store the service requests from the clients and responded data from remote servers for reducing the I/O time. Meanwhile, cache is also added on service handlers at the server side and two different caching strategies are developed for the clients and servers, respectively. The classical least-recently-used algorithm is employed at the client side, and a newly designed caching algorithm leveraging frequency-based multi-priority queues is tailored for the server side. Research shows that the transmission delay and the system response times can be significantly reduced by using the multi-level cache hierarchy for transparent computing. Based on that, an improved two-level cooperative caching strategy has recently proposed for transparent computing [136]. To evaluate the caching effectiveness and efficiency in transparent computing, a simulation framework is designed, by which the performance of multi-level cache strategies can be evaluated according to different cache configurations and replacement strategies [87].

Since the essential idea of transparent computing is to extend von Neumann architecture in network environments [177], the performance and stability of computing depend on a reliable communication network. With the development of mobile communication technologies, mobile computing devices become more and more prevailing. Therefore, it is necessary to investigate novel caching technologies suited to transparent computing in mobile network environments. In Reference [141], the authors design a block-level cache scheme according to the temporal feature of access to the server and combine with local storage access technology to optimize the block-level caching strategy by considering the limited bandwidth and communication stabilities of wireless networks.

In the mobile networks based on the all-IP cellular, the capacity of the traditional centralized mobile network cannot satisfy the demand for the explosive growth of rich multimedia contents. Recently, there are some works focusing on the issues of cache placement, including the core network, radio access network (RAN), and user devices [151]. The core network undertakes enormous traffic from RAN and user devices and exchanges inter- and intra-ISPs (Internet service providers). Therefore, it is very difficult for the core network to support massive traffic transmission. In this case, the core network is in the place where caching content is widely deployed. The research efforts on caching popular content in mobile core network have proved that the content traffic can be reduced by one third to two thirds (Reference [55] and Reference [121]). In the core network, the deployment of caching contents with the evolved packet core (EPC) draws more attention from research. For example, References [55, 154] investigate the deployed places of caching using content delivery technologies.

The backhaul links between RAN and the core network and the wireless networks between RAN and user devices become another bottleneck for mobile networks. In heterogeneous mobile networks, the multilayer architecture composing of macro-cell base stations (MBSs) and small cell base stations (SBSs) for caching and delivering content has been regarded as promising solutions for transmitting multimedia traffic [132]. MBSs need to support services for more user devices with more coverage areas, and then they can achieve a higher hit rate by data caching. SBSs with limited storage capability are more close to users in comparison with MBSs, which will be densely deployed in 5G networks [114]. Hence, user devices can retrieve contents directly from MBSs or SBSs rather than from the centric cloud. Since MBSs and SBSs are in close proximity to users, compared to the core network, the deployment of caches in the two places can reduce traffic exchange between the core network and clouds as well as traffic exchange between inter- and intra-ISPs. Therefore, some works [11, 16, 61, 116] focus on the deployment of MBS caching and SBS caching. For instance, Reference [11] propose a video-aware scheduling technique by using MBS caching to improve the video capacity and user experience. Besides, the framework relevant only to content caching at the base stations is proposed in Reference [16], which quantitatively evaluates the performance of clusters of cooperative base station caching. The device-to-device communication possesses a great many advantages, including the increase of network spectrum efficiency, offloading traffic from MBSs or SBSs, and the reduction of transmission delay [15, 137].

4.2 Contents of Cache Units

Caching at the network edge aims to alleviate the congestion of transmission between edge devices and backhaul links or clouds and to balance the tradeoff among content delivery, transmission delay, and energy consumption. Currently, increasing multimedia services are generating tremendous traffic on Internet. Moreover, a large number of users in mobile networks accesses the multimedia services by using mobile smart devices, which results in a great challenge for the multimedia content delivery and user experience [166].

It is worth noting that not all network traffic is suitable for being cached, since some un-reusable information has no need to be cached. References [121, 154] have indicated that a majority of multimedia content traffic is attributed to the duplication transmission of popular contents, while a large part of users requests for accessing to only a small portion of popular contents. Hence, existing studies focus on how to exploit the redundant request from users for reducing the duplication transmission between edge devices and the burden on the backhaul links.

From modelling popular contents perspective, the independence reference model adopted for performance analysis of web caching is utilized for most of the researches on mobile caching. In the independence reference model, contents are requested according to an independent Poisson process with the rate corresponding to the content popularity [112]. Nevertheless, the independence reference model assumes that the popularity of contents is static without considering the spatial and temporal change. In edge computing, the static model of content popularity cannot reflect the real popularity of contents due to a great deal of heterogeneity and extensive distribution of computing devices with mobility. Therefore, the dynamic analysis methods attract more attention of researchers. For instance, in Reference [28], the statistical properties of popularity distributions of requesting contents are analyzed and the opportunities utilizing the latent request for "the Long Tail" potential are discussed. The authors of Reference [142] propose a traffic model to capture the dynamics popularity of the contents requested by users, whereas these works do not consider human-driven information related to proximity to people, such as experience and preference of users as well as locations [95]. In the periphery of the networks, the information about the end-users' locations and environmental common interests should be utilized due to the dense geographical distribution of the edge devices and proximity to humans. Furthermore, in a different temporal region, the demand of users is different, namely, the popularity of the requested content may change at different temporal region. In other words, the temporal popularity of contents should be considered for caching.

4.3 Caching Strategies

It is crucial to decide what caching strategies to be adopted for caching at the edge of networks. Various caching strategies have been proposed for mobile networks, of which a portion are extended from the traditional caching strategies in wired networks by being tailored for mobile networks.

4.3.1 Consideration in Caching.

Although some conventional caching strategies with considering the content in cache units, such as the least frequently used, the least recently used, and the first-in–first-out, can alleviate the traffic congestion of transmission data with uniform size, they are not efficiently applied to the case with the variation of transmission data and delay [72]. In addition, the cache size of radio access node in mobile networks is very limited, and caching contents in radio access network are varied over time, which results in the reduction of the cache hit ratio. Considering the limited caching space in mobile cell networks, caching data at an edge server should be adjusted according to the popularity of the contents to achieve better caching efficiency. The authors of Reference [64] investigate the cache replacement strategy, which is modelled as a Markov decision process to minimize the transmission cost between edge servers in cellular networks. Then, they calculate the traffic cost for the actions of possible cache replacement in term of the previous data request and traffic between edge servers.

Recently, a few works have been written to provide a survey of caching strategies in information-centric networking [72, 170]. The emergence of information-centric networking facilitates the context-aware delivery by the deployment of in-network caching [157]. In information-centric networking, multiple superiorities have been verified by leveraging in-network caching, which includes the reduction of network traffic and the user access latency as well as the alleviation of server bottleneck. Specifically, in an IoT network, low-rate monitor and measurement data generated by large number of devices are delivered to edge servers for preprocessing. Since the IoT data possess a transient nature, it is a main challenge to cache content data at routers. Currently, the method of utilizing the in-network caching technology to mitigate the data traffic of IoT has attracted considerable attention from the research community. Reference [147] studies in-network caching related to the transient content data at routers based on a temporal data property, referred to as the *data item lifetime*. It proposes an analytical model that can well capture the balance between the multi-hop communication cost and data item freshness.

4.3.2 Proactive Caching Strategies. For emerging computing paradigms, computing services are pushed away from centralized nodes to the network edge. The perception of the specific locations and the common interests of end-users is benefited from the proximity of the edge devices. Consequently, the research efforts on content caching and delivery in the edge devices begin to consider popularity-driven caching.

Proactive caching strategies leveraging the proximity of edge computing devices have attracted large number of attentions from academia and industry. In Reference [18], the authors propose to cache proactively the popular files at both base stations and end devices during off-peak period and to leverage the correlations among social networks and D2D communication links to reduce the peak traffic demands. The similar proactive caching strategies in cloud radio access networks are proposed in Reference [37]. In addition, the popularity of contents varies with the spatial and temporal differences, and different people have different preferences on various files. The authors of Reference [11] propose a proactive caching strategy based on user preference profiles of active users in a specific cell to maximize the count of concurrent video sessions while matching the initial delay demand of the end-to-end network. Besides, since edge devices with computing and storage capabilities are close to users, many research works explore proactive caching by using machine learning to track and estimate the content timely request from end-users. In this regard, mobile operators can benefit from the exploitation of big data analysis and machine learning for the content popularity due to advantages of upcoming 5G networks. Reference [164] proposes a big data-enable architecture, in which the estimation of content popularity can be obtained by the approach of big data analysis and is used to cache contents at the base stations.

Moreover, the user mobility is the most important feature of mobile networks. Therefore, proactive caching strategies with mobility awareness have been investigated for edge caching communication networks. Reference [149] proposes a general architecture, in which the key properties of user mobility are identified to address the problem of content caching in content-centric wireless networks. Mobility-aware methodologies are further developed to model the spatial and temporal properties of mobility patterns. A mobility prediction algorithm [83] is designed to provide seamlessly content service for the users whose mobility patterns are unknown in advance. In addition, Gomes et al. [62] propose an architecture to enhance the migration of content-caches located at the edge networks by leveraging the combination of information-centric networking and the mobile follow-me cloud approach in 5G networks.

4.3.3 Clustered and Cooperative Caching Strategies. The key features of the edge computing devices include the extensive distribution and proximity to users. Hence, it is necessary to exploit these features to develop efficient content caching and distribution techniques for improving the content delivery efficiency in 5G networks.

A lot of existing clustered caching strategies aims to improve the QoS and energy efficiency of wireless networks from the caching content clustering perspective. To address the problem that the delivery of the content objects in edge caches is not accomplished between remote radio heads and the users in C-RANs, a cluster caching structure in the network edge is proposed via utilizing the centralized signal processing and distributed edge caching [181]. A similar problem is also solved by other cluster content caching paradigms [32, 74].

The cooperative caching strategy is another attractive topic in emerging computing paradigms. To transmit video contents, the authors of Reference [23] develop a lightweight algorithm to achieve both the maximum of traffic volume served from cache and the minimum of the bandwidth cost for the cooperative cache management network. The authors of Reference [75] present a strategy of content caching and delivery to coordinately alleviate the burden on backhaul and improve the content delivery efficiency by formulating the problems of cooperative content caching and content delivery as an integer-linear programming and unbalanced assignment problem. Besides, by considering the bandwidth limitation of the base stations in mobile networks, the authors of Reference [117] design a joint optimization of the caching and routing problem to maximize the content request served by the small cell base stations.

In addition, the delivery of the multimedia content depends on a reliable communication network [156, 158]. However, the spectrum resource increasingly congests in mobile networks, which limits the peak rate of the traditional cell architectures. Recently, D2D communication becomes a promising way to mitigate significantly the bottleneck of spectrum scarcity in mobile networks. Therefore, many works [60, 73] focus on the offloading of the content delivery traffic and the improvement of the network throughput by studying cache-enabled D2D communication. Different from the previous work, the authors of Reference [31] consider the interference among D2D links due to the proximity of distance between a user and the undesired transmitters. They propose a cooperation strategy to resist the interference to the D2D transmission by exploiting the caching capability of users devices.

5 SECURITY AND PRIVACY FOR EMERGING COMPUTING PARADIGMS

Security and privacy are two key factors for promoting the flourish of network computing applications. The orchestration of end, edge, and cloud makes the computing paradigms not only face the challenges inherited from cloud computing but also introduce some new challenges, such as device heterogeneity, supporting mobility, location-awareness, and low-latency services. Existing works on protecting the security and privacy for emerging computing paradigms can be briefly divided into two categories. The first is how to protect the security and privacy of the computing system itself. The second is to protect the security and privacy of the services under the computing paradigms, including service provisioning, data processing, data transmission, and data storage. In this section, the existing works are summarized from two aspects.

5.1 System-level Security and Privacy

From the security and privacy perspective, the hierarchical framework of end-edge-cloud orchestrated computing paradigms is a double-edged sword: It provides a powerful networking and computing architecture for security protection; however, it also makes the different layers of the framework vulnerable to various system-level attacks. In the following, we mainly introduce some existing works that focus on the security issues on virtualization and how to leverage the hierarchical framework for resisting threats and attacks, as well as intrusion detection.

5.1.1 Virtualization. For the emerging end-edge-cloud orchestrated computing paradigms, visualization is a key technology that can create virtual machines (VMs) to share the computation and storage resources of physical servers. However, since VMs are also vulnerable to malicious attacks and physical servers may experience system failures, it would consequently lead to unavailability of services and resources.

To increase the system dependability and robustness when a computing node has a system failure or encounters an attack, the authors of Reference [109] design a smart pre-copy live migration approach to estimate the downtime after each iteration for determining whether to proceed to the stop-and-copy stage or not. Since users usually adopt different terminals for service requesting, a user-specific trusted virtual approach is designed to adapt different security applications by dynamically instantiating in a secure place at the network edge [102]. Meanwhile, a user-defined security model is proposed to ease the security configurations transparently to heterogeneous devices. In Reference [146], the design and implementation of self-configuring honey-pots is designed to adapt the observed environment. Since in-trusted IoT devices might be interconnected toward the aggregation networks and external malware could be applied to the network, security issues may arise. The authors of Reference [145] propose edge-to-edge IoT security architecture based on network function virtualization for monitoring the current flows in IoT systems and identifying malicious flows through different anomaly detection mechanisms.

5.1.2 Threats and Attacks. To reduce the threats and attacks, some researchers intend to reinforce IoT systems from the framework perspective. By using both the centralized controller and distributed controllers, a software-defined network-based master-slave security architecture is designed in Reference [77] to enhance the security and privacy protection for the cloud users by reducing the communication distance. Moreover, an attack pattern signature on the web interface in security agent unit is defined in Reference [14] to detect persistent threats in distributed data center network and to the module on floodlight for detecting and blocking hosts with the traffic pattern in all data centers. A broad class of attacks can be detected via this method such that the security of data centers is enhanced.

5.1.3 Intrusion and Malware Detection. Intrusion and malware detection in the cloud environment has been studied for many years. However, these kinds of solutions cannot work well for the lightweight edge servers and end devices. It consequently attracts increasing research focuses on designing tailored intrusion detection mechanisms for end-edge-cloud orchestrated systems. The authors of Reference [69] develop a lightweight and distributed intrusion detection system. It is distributed in a three-layered IoT structure, where the cloud layer is to cluster primary network traffic and train its detectors, the fog layer is to analyse intrusion alerts, and the edge layer is to deploy detectors. A more general threat model is proposed in Reference [110] to design the security protection method. To detect the intrusion in private cloud, Rajendran et al. [120] propose a hybrid intrusion detection algorithm through incoming request scanning.

The detection of malware and intrusion in TC is based on its specific architecture, which provides some new clues for security mechanism design. In Reference [178], Zhang et al. propose a Meta OS platform to manage the hardware and software resources of a networked system, with the objective of providing secure and reliable computing service for end users. The Meta OS, which locates between the hardware and the OS layer, can be used to enhance the security of TC via monitoring the software and user data. A remote booting integrity service mechanism [81] is further designed to resist attacks in TC. In addition, through the streaming service execution of TC, all data obtained from the server will be eliminated from end-users after execution, and, thus, viruses have few opportunities to jeopardize those terminals. Both end devices and edge servers can monitor and manage the data streams during the transmission process [123, 175].

5.2 Service Security and Privacy

One of the biggest security challenges in IoT systems is that lightweight end devices usually have insufficient resources (e.g., computing and storage) to deploy traditional security protection mechanisms. The end-edge-cloud orchestrated computing paradigms enable edge servers and the cloud to take the security burden of end devices and build a more secure IoT system. For example, we can leverage edge servers to perform local security monitoring, threat detection, and threat protection on behalf of the endpoints [42]. However, when we consider the services provided by these computing paradigms, we should think about the following questions: whether the devices should be trusted; how to preserve the data privacy when edge servers provide services; how to guarantee that the service can be accessed by authorized entities. In the literature, there have been a large number of works proposed to answer these questions. We briefly introduce them from the following three perspectives.

5.2.1 Authentication and Trust. When assigning an IoT device to an edge server, authenticating the identity of the owner to a specific edge server needs to be considered [122]. To reduce the authentication burden on IoT devices, Reference [135] designs an electroencephalography authentication system by performing machine learning algorithms on an edge server (i.e., laptop). In this system, IoT devices capture the raw electroencephalography signals and sends them to a smart phone via Bluetooth. On a smart phone, the system interface application receives electroencephalography signals and sends them to a laptop. The laptop uses machine learning algorithms to make identification/authentication. To realize efficient authentication, the authors of Reference [71] develop a secure and efficient approach for fog computing, which allows fog users to mutually authenticate with each other under the authority of a cloud service provider. The proposed approach does not require a fog user to be included in a public key infrastructure. The fog user only needs to store one master secret key in the registration phase once to mutually authenticate with any other fog server. Moreover, the authors of Reference [51] propose an efficient and responsive mutual authentication scheme by using elliptic curve cryptography on edge servers to protect smart grid system. Considering the scenario without a trust third-party in an IoT network, Reference [53] establishes trusted identities in disconnected edge environments by combining identity-based cryptography with secure key generation and exchange mechanisms. To integrate different authentication protocols on heterogeneous IoT devices, Reference [128] proposes a common identity and authentication scheme, which is able to resist many typical attacks in IoT scenarios, e.g., replay, man-in-the-middle, and masquerade attacks.

5.2.2 Privacy Preservation. As the edge devices are widely deployed to process the generated data from IoT devices (smart grid meters in home area networks, body-sensors that collect health information) and provide network services (such as data storage and data sharing), a privacy leakage problem may arise during data processing provided at the network edge. As a classical technique, homomorphic encryption has been applied to providing privacy-preserving aggregation on encrypted data at the network edge [161]. In Reference [148], the authors propose an anonymous and secure aggregation scheme by which edge servers can help IoT devices upload their sensed data to a service provider. The proposed scheme can well preserve the identities of devices with pseudonyms and guarantee the data secrecy based on homomorphic encryption, simultaneously. Lu et al. [92] present a privacy-preserving and efficient aggregation scheme for communication in smart grid. It structures multi-dimensional data by utilizing a super-increasing sequence and encrypts the structured data with homomorphic cryptography techniques. A homomorphic function uses the encrypted data of the smart meters as input and generates the aggregated and encrypted results.

Reference [91] proposes a privacy-preserving data aggregation scheme to aggregate hybrid IoT data and filter injected false data at the network edge. A fog-based de-duplicated spatial crowdsourcing framework is designed to allocate task and sense data deduplication security for fog-assisted crowdsourcing system [106]. To realize secure duplication in crowdsourcing environments for storage, the authors of Reference [79] propose a hybrid secure deduplication client-server protocol for untrustworthy fog storage scenarios. The authors of Reference [19] propose a certificateless aggregate signcryption scheme for preserving privacy in a vehicular crowdsensing system. In addition, Reference [100] proposes an efficient data sharing scheme to securely share data among edge devices in a cloud-assisted IoT system.

5.2.3 Access Control. As an important research topic in IoT systems, access control has also attracted increasing attention under the hierarchical architecture of emerging computing paradigms [140]. In the area of e-healthcare systems, Reference [13] integrates the access security broker to fog nodes and proposes an attribute-based access control approach to realize fast response for latency-sensitive e-healthcare applications and to prevent unauthorized access to health information. For providing secure services for heterogeneous resources, a policy-based access control mechanism is developed to achieve secure collaboration and interoperability in a distributed manner [52]. A holistic solution is exploited to achieve multi-factor access control for edge servers [56]. This solution enables layered security for both of the extrinsic parties and the storage service providers and is robust in resisting the colluding attacks by a (small) fraction of service providers.

Since IoT devices (such as wearable devices and wireless sensors) are usually resource limited, they lack powerful computation ability and face the security challenges to protect themselves or provide complex cryptographic algorithms to preserve their generated data. Researchers integrate edge computing to provide complementary security solutions for IoT devices to reduce the computation burden. To decrease the computational cost of decryption, Reference [182] introduces the attribute-based encryption with outsourced decryption on edge servers to offload the large amount of computation. Reference [101] designs a privacy-preserving system to enable the cloud users to enhance the privacy and security of sensitive personal data. By combining attribute-based and role-based access control models, a new approach is proposed to preserve the data privacy and satisfy both of mandatory and discretionary access control requirements.

6 FUTURE RESEARCH DIRECTIONS

By fully exploiting the resources of end devices and the nearby devices/infrastructures, the emerging network computing paradigms are paving the road toward a more scalable and intelligent IoT era. Although existing studies provide comprehensive investigations on different network computing paradigms from various aspects, end-edge-cloud computing is still in its infancy and needs continuous research efforts to fuel the explosion of more intelligent IoT applications. In this section, we briefly outline some future research directions to foster further research.

6.1 Lightweight Virtualization for Network Computing

The dramatically increasing IoT devices have brought new challenges to fully exploit the heterogeneous resources associated to them. The lightweight hardware and high heterogeneity of IoT devices also bring various customized operating systems and software, hindering the flexibility of IoT services and service sharing among different IoT applications. Powered by the advanced network computing technologies, device virtualization solutions have shown the potentials for providing desired tradeoff between flexibility and performance. Redesigning the virtualization methods for lightweight IoT and edge devices becomes a growing trend in the research field of virtualization.

Some emerging technologies, such as Docker and LXC containers, have been already successfully deployed on single-board computers to provide virtualized instances with an efficient overhead.

As a representative of operating-system-level virtualization technologies, containerization has refashioned the world of software development by introducing flexibility and new way of managing and distributing software. In Reference [126], the authors analyze the relation between the lightweight virtualization and edge computing. They have also answered how to effectively combine these two technologies and pointed out some classical application scenarios. Morabito et al. [103] propose two container-based IoT service provisioning frameworks and compare their performance based on a real IoT testbed. The one is for the case where two cooperating devices interact directly, and the other is for the case where a manager can supervise the operations among cooperating devices. Although existing works provide some valuable preliminary results on applying container-based virtualization solutions in IoT scenarios, there are still many technical challenges in developing more reliable and robust lightweight virtualization technologies. For example, how to orchestrate the edge resources and elements, how to monitor the performance and status of different service instances, how to provide security and privacy guarantees for virtualized applications, as well as how to achieve efficient service migration among different containers. These remaining challenges leave a huge space for researchers to study more mature and advanced lightweight virtualization technologies for network computing systems.

6.2 Block-streaming Service Loading

Besides running services on edge servers by lightweight virtualization (like Software-as-a-Service in cloud computing), dynamically loading on-demand services and executing them on IoT devices is an alternative solution for dynamic service provisioning in the IoT era. In 2003, the authors of Reference [118] propose a stream way to execute the software on devices. In such a way, the software can be executed when it is being streamed without waiting for completing the whole process of downloading, decompression, installation, and reconfiguration. A streaming approach can reduce the experienced application loading time of users, since the application can start running once the first executable unit has been loaded into the memory. Recently, He et al. [67] propose a new cloud service model based on transparent computing, named Block-stream as a service (BaaS), for IoT devices. The BaaS model provides a framework for lightweight IoT devices to load and execute part of service codes on demand, by fully exploiting the advantages of transparent computing in terms of cross-platform and dynamic service provisioning. Inspired by the BaaS model, Peng et al. [115] propose a transparent computing-based block-streaming application execution scheme, called BOAT, and implement it on a lightweight wearable platform. It can remotely load a part of necessary application codes from edge servers and execute the codes on IoT devices locally. Their experimental results demonstrate that BOAT can achieve significant improvements in terms of service loading delay and energy consumption when compared to the traditional Software-as-a-Service approach. Some similar concepts, like microservices [143], have also been proposed to improve the modularity, distribution, elasticity, and robustness of service provisioning for IoT devices by leveraging the capability of network computing.

Although block-streaming service loading has shown the potential to revolutionize the service provisioning way for lightweight IoT devices, continuous efforts should be devoted in this research direction to address some open challenges. The bandwidth-limited and unstable wireless communications may make dynamic service loading intermittent and consequently lead to poor quality of experience (QoE). How to design efficient service prefetching mechanisms becomes an important research direction to improve the QoE of users under the stochastic wireless communication environment. Moreover, since the whole service may be divided into different parts to response the service request of IoT devices, the integrity check and verification of the service should be

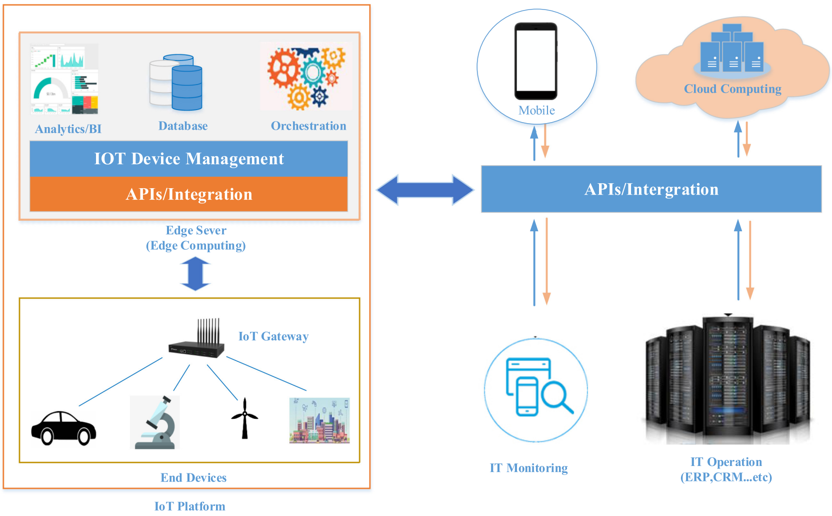


Fig. 7. APIs for end-edge-cloud orchestrated computing systems.

carefully redesigned to address the security risk. In addition, if the software of IoT services is programmed in a modular way, then the efficiency of block-streaming service loading can be further enhanced. It means that standards or rules can be made to regulate the IoT software programs to guarantee the modularity of IoT services.

6.3 API Standards and Development for End-Edge-Cloud Orchestrated Computing

To standardize and facilitate application development for end-edge-cloud orchestrated computing systems, investigating the Application Programming Interface (API) standards and developing flexible APIs have been regarded as a very important research direction. A well-defined API consists of subroutine definitions, communication protocols, and tools for building software connections, which can provide flexibility and seamless connectivity for different kinds of devices and authorize other pieces of software to control the functionality of an application or a service. Figure 7 illustrates how APIs work for the orchestration of end-edge-cloud systems [6]. As shown in the figure, the APIs developed for edge servers and the cloud should be carefully designed and coordinated by considering their different resources and capabilities. However, due to the complexity and dynamics caused by the orchestration of end-edge-cloud resources, the design, management, and integration of APIs are facing great challenges. The first one is, from the communication perspective, how to provide flexible and seamless connectivity for highly mobile users to distributed edge servers and edge servers to the cloud under the hybrid accessing technologies. The second one is, from the service perspective, how to design suitable APIs and management/integration techniques that can fully exploit and coordinated the differentiated resources and capabilities of devices for managing service provisioning and application development. Especially when microservices are becoming the main trend in the IoT era, dividing applications into feature components and microservices drives API management and integration more significant than ever. Components and services that may be only parts of a single application need to be well integrated for working in a coordinated way and to deliver the requested capabilities for users. Thus, more research efforts should be devoted to the standardization and development of APIs for end-edge-cloud orchestrated computing.

6.4 Flexible Networked Operating System Design for Heterogeneous IoT Devices

The types of IoT devices range from ultra lightweight RFID tags to some powerful smart phones and vehicles. The hardware heterogeneity of IoT devices also leads to a wide variety of IoT operating systems in the community. For most of lightweight IoT devices, embedded realtime OSes, such as TinyOS and Contiki, are adopted to provide specific functionalities with strict delay constraints. But for the powerful IoT devices, general OSes, such as Android and iOS, are widely applied to support resource sharing and multi-task processing. Diverse OSes cause an inevitable limitation that IoT services cannot be shared among different IoT devices. And in most cases, even with the same operating system kernel, the OSes developed by different vendors cannot support cross-platform service sharing. Therefore, designing a flexible operating system that can be well applied on heterogeneous IoT devices attracts increasing attention from both of the academia and industry. For example, Google is developing a new IoT OS, named Fuchsia, which is based on a new kernel named Magenta. It aims to provide flexible deployment on heterogeneous IoT devices by using a micro-kernel and a large set of incremental services and drivers. IoT devices can choose to add functionalities on the micro-kernel according to their capabilities and requirements, such that cross-platform services can be well supported by any kind of devices. However, with the development of network computing technologies, the network-connected IoT devices can be regarded as an integrated system. We can also leverage the power of network computing to change the traditional way of OS design. Flexible networked OS, where OS kernels or functionalities are distributed over different connected devices, may be a new trend for IoT operating system design.

6.5 Lightweight and Distributed Machine Learning for Realtime Intelligence

Recent advances in machine learning, especially in deep learning, have surged the development of intelligent IoT applications, such as mobile face recognition, video analytics, and autonomous path planning drones. These kinds of intelligent applications rely on computationally intensive machine learning algorithms and require real-time processing. Due to the hardware limitations, however, mobile and IoT devices are facing great challenges to fulfill stringent QoS requirements of intelligent applications. For example, video analytics, which usually involves complex deep neural networks, far exceeds the computing capabilities of the end devices. Meanwhile, offloading the videos to the cloud server incurs unbearable burdens on the core network and produces unacceptable delay, which consequently hinders the efficiency of video analytics. Leveraging network computing to directly perform machine learning and data analysis at mobile terminals or the network edge becomes a promising way to provide computation augmenting and timely intelligent services for mobile and IoT devices. This has also been regarded as the “killer application” of edge computing.

However, most IoT devices and edge servers are not capable enough to support the machine learning and data analytical algorithms conventionally deployed in the resource-extensive cloud. To well support intelligent mobile and IoT applications, therefore, there is a significant need for exploring the implementation of lightweight and distributed machine learning models at IoT devices and the network edge, studying efficient resource allocation algorithms for real-time machine learning among devices, edge servers, and the cloud, as well as designing collaborative and distributed data analyzing architectures for network computing paradigms. Recently, Google proposed a decentralized machine learning approach, named federated learning, to train the data distributed among lightweight IoT devices then to merge the aggregated locally computed updates to one shared model [78]. It introduces a federated learning algorithm that combines local stochastic gradient descent training on each client within relatively few communication rounds where the central cloud server performs model averaging. Following the idea of distributed learning, Hardy et al. [45] propose AdaComp, a distributed deep learning algorithm running on edge devices, which can compress the edge updates to the model on cloud servers. These existing works have pointed out a potential research direction for implementing lightweight and distributed machine learning algorithms, with the aim of achieving efficient and timely data training and analysis of IoT devices and edge servers, and reducing the communication rounds between the edge side and the cloud side. In such a way, the privacy of personal data can also be well preserved, since the data traffic originally transmitted to the cloud for model training and analysis can be significantly reduced.

6.6 Advanced Security and Privacy Preservation Techniques for Network Computing

Since lightweight IoT devices are generally limited by computing capability, storage, and energy supply, traditional security solutions, e.g., complex cryptographic solutions, may not be applicable for IoT devices. Thus, how to leverage network computing to offload the security protection burden of IoT platforms becomes a potential and promising research direction. As we discussed in the previous section, there have been many existing works focused on offloading the authentication and encryption burden from lightweight IoT devices to edge servers. These solutions provide some valuable insights to address the security and privacy problems in current IoT applications.

However, compared to the resourceful cloud servers, the resource-constrained edge servers are still vulnerable to different kinds of attacks. It consequently is necessary that future security techniques should fully leverage the capabilities of edge servers to provide enhanced security protection for IoT devices and also can guarantee the security of the edge servers. Thus, how to efficiently utilize the three-tire framework, i.e., end-edge-cloud, to collaboratively achieve advanced security and privacy preservation for the whole IoT platform becomes an important research direction for future studies. Moreover, for ultra-light-weight IoT devices, e.g., RFID tags, even very lightweight security approaches cannot be efficiently implemented. Since such kinds of IoT devices will take a significant portion of "connected things" and face great security and privacy leakage risks in the IoT era, more attention should be paid to leveraging network computing for addressing this challenge.

7 CONCLUSION

In this survey, we have provided an overview and comparison of the emerging network computing paradigms, including transparent computing, mobile edge computing, fog computing, and cloudlet computing. The common feature of the computing paradigms is to orchestrate the computing and storage resources of end devices, edge servers, and the cloud to improve the performance of hierarchical IoT systems. From this perspective, we have presented a comprehensive survey on some hot research issues of the computing paradigms, in terms of computation offloading, caching, as well as security and privacy. Finally, we have discussed some potential future directions for emerging and evolving research.

REFERENCES

[1] Mobile Edge Computing Introductory Technical White Paper. 2014. Retrieved from https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf.

[2] Mobile-Edge Computing (MEC); Service Scenarios. 2015. Retrieved from http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf.

[3] Edge computing prepares for a multi-access future. 2016. Retrieved from http://www.telecomtv.com/articles/mec/edge-computing-prepares-for-a-multi-access-future-13986/.

[4] Mobile-Edge Computing (MEC); Technical Requirements. 2016. Retrieved from http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf.

[5] Mobile-Edge Computing (MEC); Terminology. 2016. Retrieved from http://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/01.01.01_60/gs_MEC001v010101p.pdf.

[6] APIs for IOT and FOG computing. 2017. Retrieve from https://www.vanrish.com/blog/2017/12/17/apis-for-iot-and-fog-computing/.

[7] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021. 2017. Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[8] Cloud Adoption Is Growing But Forecasts Differ on How Much. 2017. Retrieved from http://fortune.com/2017/02/22/cloud-growth-forecast-gartner/.

[9] Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines. 2017. Retrieved from http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/006/01.01.01_60/gs_MEC-IEG006v010101p.pdf.

[10] OpenFog Reference Architecture for Fog Computing. 2017. Retrieved from https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf.

[11] Hasti Ahlehagh and Sujit Dey. 2014. Video-aware scheduling and caching in the radio access network. *IEEE/ACM Trans. Netw.* 22, 5 (2014), 1444–1462.

[12] Arif Ahmed and Ejaz Ahmed. 2016. A survey on mobile edge computing. In *Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO'16).* 1–8. DOI: https://doi.org/10.1109/ISCO.2016.7727082

[13] Mohammed Alshiky Aisha, M. Buhari Seyed, and Barnawi Ahmed. 2017. Attribute-based access control (ABAC) for EHR in fog computing environment. *Int. J. Cloud Comput.: Serv. Arch.* 7, 1 (2017), 9–16.

[14] Moustafa Ammar, Mohamed Rizk, Ayman Abdel-Hamid, and Ahmed K. Aboul-Seoud. 2016. A framework for security enhancement in SDN-based datacenters. In *Proceedings of the IFIP International Conference on New Technologies, Mobility and Security (NTMS'16).* IEEE, 1–4.

[15] Arash Asadi, Qing Wang, and Vincenzo Mancuso. 2014. A survey on device-to-device communication in cellular networks. *IEEE Commun. Surv. Tutor.* 16, 4 (2014), 1801–1819.

[16] Bahar Azari, Osvaldo Simeone, Umberto Spagnolini, and Antonia M. Tulino. 2016. Hypergraph-based analysis of clustered co-operative beamforming with application to edge caching. *IEEE Wireless Commun. Lett.* 5, 1 (2016), 84–87.

[17] Bo Bai, Li Wang, Zhu Han, Wei Chen, and Tommy Svensson. 2016. Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework. *IEEE Wireless Commun.* 23, 4 (2016), 74–81.

[18] Ejder Bastug, Mehdi Bennis, and Mérouane Debbah. 2014. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Commun. Mag.* 52, 8 (2014), 82–89.

[19] Sultan Basudan, Xiaodong Lin, and Karthik Sankaranarayanan. 2017. A privacy-preserving vehicular crowdsensing based road surface condition monitoring system using fog computing. *IEEE IoT J.* 4, 3 (2017), 772–782.

[20] Suzhi Bi and Ying Jun Zhang. 2018. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. *IEEE Trans. Wireless Commun.* 17, 6 (Jun. 2018), 4177–4190. DOI: https://doi.org/10.1109/TWC.2018.2821664

[21] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. 2014. *Fog Computing: A Platform for Internet of Things and Analytics.* Springer International Publishing, Cham, 169–186. DOI: https://doi.org/10.1007/978-3-319-05029-4_7

[22] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the Internet of Things. In *Proceedings of the Workshop on Mobile Cloud Computing (MCC'12).* ACM, New York, NY, 13–16.

[23] Sem Borst, Varun Gupta, and Anwar Walid. 2010. Distributed caching algorithms for content distribution networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'10).* IEEE, 1–9.

[24] Piotr Borylo, Artur Lason, Jacek Rzasa, Andrzej Szymanski, and Andrzej Jajszczyk. 2016. Energy-aware fog and cloud interplay supported by wide area software defined networking. In *Proceedings of the IEEE International Conference on Communications (ICC'16).* IEEE, 1–7.

[25] A. Bourdena, C. X. Mavromoustakis, G. Mastorakis, J. J. P. C. Rodrigues, and C. Dobre. 2015. Using socio-spatial context in mobile cloud process offloading for energy conservation in wireless devices. *IEEE Trans. Cloud Comput.* 1 (2015).

[26] Shiwei Cao, Xiaofeng Tao, Yanzhao Hou, and Qimei Cui. 2015. An energy-optimal offloading algorithm of mobile computing based on HetNets. In *Proceedings of the IEEE International Conference on Connected Vehicles and Expo (ICCVE'15).* IEEE, 254–258.

[27] Sami Yangui Roch H. Glitho Monique J. Morrow Carla Mouradian, Diala Naboulsi and Paul A. Polakos. 2018. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* 20, 1 (2018), 416–464.

[28] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. 2009. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.* 17, 5 (2009), 1357–1370.

[29] Omar Chakroun and Soumaya Cherkaoui. 2016. Resource allocation for delay sensitive applications in mobile cloud computing. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN'16).* IEEE, 615–618.

[30] Binqiang Chen, Chenyang Yang, and Gang Wang. 2016. Cooperative device-to-device communications with caching. In *Proceedings of the IEEE Vehicular Technology Conference (VTC'16).* IEEE, 1–5.

[31] Binqiang Chen, Chenyang Yang, and Gang Wang. 2017. High throughput opportunistic cooperative device-to-device communications with caching. *IEEE Trans. Vehic. Technol.* 66, 8 (2017), 7527–7539.

[32] Di Chen, Stephan Schedler, and Volker Kuehn. 2016. Backhaul traffic balancing and dynamic content-centric clustering for the downlink of fog radio access network. In *Proceedings of the IEEE SPAWC*. IEEE, 1–5.

[33] Guangyu Chen, B.-T. Kang, Mahmut Kandemir, Narayanan Vijaykrishnan, Mary Jane Irwin, and Rajarathnam Chandramouli. 2004. Studying energy trade offs in offloading computation/compilation in java-enabled mobile devices. *IEEE Trans. Parallel Distrib. Syst.* 15, 9 (2004), 795–809.

[34] Lixing Chen, Sheng Zhou, and Jie Xu. 2018. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. *IEEE/ACM Trans. Netw.* 26, 4 (2018), 1619–1632. DOI: https://doi.org/10.1109/TNET.2018.2841758

[35] Min Chen, Yixue Hao, Chin-Feng Lai, Di Wu, Yong Li, and Kai Hwang. 2018. Opportunistic task scheduling over co-located clouds in mobile environment. *IEEE Trans. Serv. Comput.* 11, 3 (2018), 549–561.

[36] M. Chen, Y. Hao, Y. Li, C. Lai, and D. Wu. 2015. On the computation offloading at ad hoc cloudlet: Architecture and service modes. *IEEE Commun. Mag.* 53, 6 (Jun. 2015), 18–24. DOI: https://doi.org/10.1109/MCOM.2015.7120041

[37] Mingzhe Chen, Walid Saad, Changchuan Yin, and Mérouane Debbah. 2017. Echo state networks for proactive caching in cloud-based radio access networks with mobile users. *IEEE Trans. Wireless Commun.* 16, 6 (2017), 3520–3535.

[38] Mingkai Chen, Lei Wang, and Jianxin Chen. 2016. Users-media cloud assisted D2D communications for distributed caching underlaying cellular network. *Chin. Commun.* 13, 8 (2016), 13–23.

[39] Meng-Hsi Chen, Min Dong, and Ben Liang. 2016. Joint offloading decision and resource allocation for mobile cloud with computing access point. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'16)*. IEEE, 3516–3520.

[40] Meng-Hsi Chen, Ben Liang, and Min Dong. 2015. A semidefinite relaxation approach to mobile cloud offloading with computing access point. In *Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC'15)*. IEEE, 186–190.

[41] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. 2016. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* 24, 5 (2016), 2795–2808.

[42] Mung Chiang, Sangtae Ha, I. Chih-Lin, Fulvio Risso, and Tao Zhang. 2017. Clarifying fog computing and networking: 10 questions and answers. *IEEE Commun. Mag.* 55, 4 (2017), 18–20.

[43] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of the the 6th Conference on Computer Systems*. ACM, 301–314.

[44] Byung-Gon Chun and Petros Maniatis. 2010. Dynamically partitioning applications between weak devices and clouds. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*. ACM, 7.

[45] Hardy Corentin, Le Merrer Erwan, and Sericola Bruno. 2017. Distributed deep learning on edge-devices: Feasibility via adaptive compression. In *Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA'17)*. IEEE, 1–8.

[46] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. 2010. MAUI: Making smartphones last longer with code offload. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (Mobisys'10)*. ACM, 49–62.

[47] Amir Vahid Dastjerdi and Rajkumar Buyya. 2016. Fog computing: Helping the Internet of Things realize its potential. *Computer* 49, 8 (2016), 112–116.

[48] Maofei Deng, Hui Tian, and Bo Fan. 2016. Fine-granularity based application offloading policy in cloud-enhanced small cell networks. In *Proceedings of the IEEE International Conference on Communications (ICC'16)*. IEEE, 638–643.

[49] Haichuan Ding and Yuguang Fang. 2018. Virtual infrastructure at traffic lights: Vehicular temporary storage assisted data transportation at signalized intersections. *IEEE Trans. Vehic. Technol.* 67, 12 (2018), 12452–12456.

[50] Haichuan Ding, Yuanxiong Guo, Xuanheng Li, and Yuguang Fang. 2018. Beef up the edge: Spectrum-aware placement of edge computing services for the Internet of Things. *IEEE Trans. Mobile Comput.* (2018). DOI: https://doi.org/10.1109/TMC.2018.2883952

[51] Abebe Abeshu Diro, Naveen Chilamkurti, and Neeraj Kumar. 2017. Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing. *Mobile Netw. Appl.* 22, 5 (2017), 848–858.

[52] Clinton Dsouza, Gail-Joon Ahn, and Marthony Taguinod. 2014. Policy-driven security management for fog computing: Preliminary framework and a case study. In *Proceedings of the IEEE International Conference on Information Reuse and Information (IRI'14)*. IEEE, 16–23.

[53] Sebastián Echeverría, Dan Klinedinst, Keegan Williams, and Grace A. Lewis. 2016. Establishing trusted identities in disconnected edge environments. In *Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC'16)*. IEEE, 51–63.

[54] Khalid Elgazzar, Patrick Martin, and Hossam S. Hassanein. 2016. Cloud-assisted computation offloading to support mobile services. *IEEE Trans. Cloud Comput.* 4, 3 (2016), 279–292.

[55] Jeffrey Erman, Alexandre Gerber, Mohammad Hajiaghayi, Dan Pei, Subhabrata Sen, and Oliver Spatscheck. 2011. To cache or not to cache: The 3G case. *IEEE Internet Comput.* 15, 2 (2011), 27–34.

[56] Ertem Esiner and Anwitaman Datta. 2016. Layered security for storage at the edge: On decentralized multi-factor access control. In *Proceedings of the ACM International Conference on Distributed Computing and Networking (ICDCN'16)*. ACM, 9.

[57] Jose Oscar Fajardo, Ianire Taboada, and Fidel Liberal. 2015. Radio-aware service-level scheduling to minimize downlink traffic delay through mobile edge computing. In *Proceedings of the 7th EAI International Conference on Mobile Networks and Management*. Springer, 121–134.

[58] Yuan Gao, Yaoxue Zhang, and Yuezhi Zhou. 2012. A cache management strategy for transparent computing storage system. In *Proceedings of the International Conference on New Trends in Computing Sciences (ICTCS'12)*. Springer, 651–658.

[59] Ioana Giurgiu, Oriana Riva, Dejan Juric, Ivan Krivulev, and Gustavo Alonso. 2009. Calling the cloud: Enabling mobile phones as interfaces to cloud applications. In *Proceedings of the International Middleware Conference (Middleware'09)*. Springer-Verlag New York, 5.

[60] Negin Golrezaei, Parisa Mansourifard, Andreas F Molisch, and Alexandros G Dimakis. 2014. Base-station assisted device-to-device communications for high-throughput wireless video networks. *IEEE Trans. Wireless Commun.* 13, 7 (2014), 3665–3676.

[61] Negin Golrezaei, Karthikeyan Shanmugam, Alexandros G. Dimakis, Andreas F. Molisch, and Giuseppe Caire. 2012. Femtocaching: Wireless video content delivery through distributed caching helpers. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'12)*. IEEE, 1107–1115.

[62] Andre S. Gomes, Bruno Sousa, David Palma, Vitor Fonseca, Zhongliang Zhao, Edmundo Monteiro, Torsten Braun, Paulo Simoes, and Luis Cordeiro. 2017. Edge caching with mobility prediction in virtualized LTE mobile networks. *Future Gener. Comput. Syst.* 70, 5 (2017), 148–162.

[63] Mohammad Goudarzi, Zeinab Movahedi, and Masoud Nazari. 2016. Mobile cloud computing: A multisite computation offloading. In *Proceedings of the IEEE International Conference on Imaging Systems and Techniques (IST'16)*. IEEE, 660–665.

[64] Jingxiong Gu, Wei Wang, Aiping Huang, Hangguan Shan, and Zhaoyang Zhang. 2014. Distributed cache replacement for caching-enable base stations in cellular networks. In *Proceedings of the IEEE International Conference on Communications (ICC'14)*. IEEE, 2648–2653.

[65] Hongzhi Guo and Jiajia Liu. 2018. Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks. *IEEE Trans. Vehic. Technol.* 67, 5 (May 2018), 4514–4526.

[66] Kiryong Ha and Mahadev Satyanarayanan. 2015. *OpenStack++ for Cloudlet Deployment*. Technical Report. http://elijah.cs.cmu.edu/DOCS/CMU-CS-15-113.pdf

[67] J. He, Y. Zhang, J. Lu, M. Wu, and F. Huang. 2018. Block-stream as a service: A more secure, nimble, and dynamically balanced cloud service model for ambient computing. *IEEE Netw.* 32, 1 (2018), 126–132.

[68] Dinh T. Hoang, Chonho Lee, Dusit Niyato, and Ping Wang. 2013. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Commun. Mobile Comput.* 13, 18 (2013), 1587–1611.

[69] Farhoud Hosseinpour, Payam Vahdani Amoli, Juha Plosila, Timo Hämäläinen, and Hannu Tenhunen. 2016. An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach. *Int. J. Dig. Content Technol. Appl.* 10, 5 (2016), 34–46.

[70] Yunchao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. 2015. ETSI. White Paper No. 11. Mobile Edge Computing a Key Technology Towards 5G [S]. DOI: https://doi.org/10.1007/978-3-319-05029-4_7

[71] Maged Ibrahim. 2016. Octopus: An edge-fog mutual authentication scheme. *Int. J. Netw. Secur.* 18, 6 (2016), 1089–1101.

[72] Andriana Ioannou and Stefan Weber. 2016. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Commun. Surv. Tutor.* 18, 4 (2016), 2847–2886.

[73] Mingyue Ji, Giuseppe Caire, and Andreas F. Molisch. 2016. Wireless device-to-device caching networks: Basic principles and system performance. *IEEE J. Select. Areas Commun.* 34, 1 (2016), 176–189.

[74] Shiwei Jia, Yuan Ai, Zhongyuan Zhao, Mugen Peng, and Chunjing Hu. 2016. Hierarchical content caching in fog radio access networks: Ergodic rate and transmit latency. *Chin. Commun.* 13, 12 (2016), 1–14.

[75] Wei Jiang, Gang Feng, and Shuang Qin. 2017. Optimal cooperative content caching and delivery policy for heterogeneous cellular networks. *IEEE Trans. Mobile Comput.* 16, 5 (2017), 1382–1393.

[76] Sladana Josĭlo and György Dán. 2019. Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks. *IEEE Trans. Mobile Comput.* 18, 1 (2019), 207–220. DOI: https://doi.org/10.1109/TMC.2018.2829874

[77] Sarang Kahvazadeh, Vitor B. Souza, Xavi Masip-Bruin, Eva Marn-Tordera, Jordi Garcia, and Rodrigo Diaz. 2017. Securing combined fog-to-cloud system through SDN approach. In *Proceedings of the Workshop on CrossCloud Infrastructures & Platforms (CrossCloud'17)*. ACM, 2.

[78] Jakub Koneáý, H. Brendan, McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. arXiv:1610.02527v1 (10 2016).

[79] Dongyoung Koo, Youngjoo Shin, Joobeom Yun, and Junbeom Hur. 2016. A hybrid deduplication for secure and efficient data outsourcing in fog computing. In *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom'16)*. IEEE, 285–293.

[80] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. 2012. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'12)*. IEEE, 945–953.

[81] Wenyuan Kuang, Yaoxue Zhang, Yuezhi Zhou, and Huajie Yang. 2007. RBIS: Security enhancement for MRBP and MRBP2 using integrity check. *J. Chin. Comput. Syst.* 28, 2 (2007), 251–254.

[82] D. Van Le and C. Tham. 2018. A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*. 760–765.

[83] Hongjia Li and Dan Hu. 2016. Mobility prediction based seamless RAN-cache handover in HetNet. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC'16)*. IEEE, 1–7.

[84] Xue Lin, Yanzhi Wang, Qing Xie, and Massoud Pedram. 2015. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment. *IEEE Trans. Serv. Comput.* 8, 2 (2015), 175–186.

[85] Ying-Dar Lin, Edward T-H Chu, Yuan-Cheng Lai, and Ting-Jun Huang. 2015. Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds. *IEEE Syst. J.* 9, 2 (2015), 393–405.

[86] Juan Liu, Yuyi Mao, Jun Zhang, and Khaled B. Letaief. 2016. Delay-optimal computation task scheduling for mobile-edge computing systems. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT'16)*. IEEE, 1451–1455.

[87] Jinzhao Liu, Yuezhi Zhou, and Di Zhang. 2016. TranSim: A simulation framework for cache-enabled transparent computing systems. *IEEE Trans. Comput.* 65, 10 (2016), 3171–3183.

[88] Liqing Liu, Zheng Chang, and Xijuan Guo. 2018. Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices. *IEEE IoT J.* 5, 3 (Jun. 2018), 1869–1879.

[89] Yanchen Liu, Myung J Lee, and Yanyan Zheng. 2016. Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. *IEEE Trans. Mobile Comput.* 15, 10 (2016), 2398–2410.

[90] Garcia Pedro Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. Edge-centric computing: Vision and challenges. *SIGCOMM Comput. Commun. Rev.* 45, 5 (2015), 37–42.

[91] Rongxing Lu, Kevin Heung, Arash Habibi Lashkari, and Ali A. Ghorbani. 2017. A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT. *IEEE Access* 5 (2017), 3302–3312.

[92] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuemin Shen. 2012. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Trans. Parallel Distrib. Syst.* 23, 9 (2012), 1621–1631.

[93] Pavel Mach and Zdenek Becvar. 2016. Cloud-aware power control for real-time application offloading in mobile edge computing. *Trans. Emerg. Telecommun. Technol.* 27, 5 (2016), 648–661.

[94] S. Eman Mahmoodi, R. N. Uma, and K. P Subbalakshmi. 2019. Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Trans. Cloud Comput.* 7, 2 (2019), 301–313.

[95] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. 2017. Mobile edge computing: Survey and research outlook. *arXiv preprint* arXiv:1701.01090 (2017).

[96] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* 19, 4 (2017), 2322–2358.

[97] Yuyi Mao, Jun Zhang, and Khaled B. Letaief. 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Sel. Areas Commun.* 34, 12 (2016), 3590–3605.

[98] Yuyi Mao, Jun Zhang, S. H. Song, and Khaled B. Letaief. 2016. Power-delay tradeoff in multi-user mobile-edge computing systems. *arXiv preprint* arXiv:1609.06027 (2016).

[99] Yuyi Mao, Jun Zhang, S. H. Song, and Khaled B. Letaief. 2017. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Trans. Wireless Commun.* 16, 9 (2017), 5994–6009.

[100] Muhammad Baqer Mollah, Md Abul Kalam Azad, and Athanasios Vasilakos. 2017. Secure data sharing and searching at the edge of cloud-assisted Internet of Things. *IEEE Cloud Comput.* 4, 1 (2017), 34–42.

[101] Ei Ei Mon and Thinn Thu Naing. 2011. The privacy-aware access control system using attribute-and role-based access control in private cloud. In *Proceedings of the IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT'11)*. IEEE, 447–451.

[102] Diego Montero, Marcelo Yannuzzi, Adrian Shaw, Ludovic Jacquin, Antonio Pastor, Rene Serral-Gracia, Antonio Lioy, Fulvio Risso, Cataldo Basile, Roberto Sassu, et al. 2015. Virtualized security at the network edge: A user-centric approach. *IEEE Commun. Mag.* 53, 4 (2015), 176–186.

[103] R. Morabito, I. Farris, A. Iera, and T. Taleb. 2017. Evaluating performance of containerized IoT services for clustered devices at the network edge. *IEEE IoT J.* 4, 4 (2017), 1019–1030.

[104] Chiang Mung, Ha Sangtae, I Chih-lin, Risso Fulvio, and Zhang Tao. 2017. Clarifying fog computing and networking: 10 questions and answers. *IEEE Commun. Mag.* 55, 4 (Apr. 2017), 18–20. DOI: https://doi.org/10.1109/MCOM.2017.7901470

[105] Olga Muñoz, Antonio Pascual Iserte, Josep Vidal, and Marc Molina. 2014. Energy-latency trade-off for multiuser wireless computation offloading. In *Proceedings of the IEEE Wireless Communications and Networking Conference Workshop (WCNCW'14)*. IEEE, 29–33.

[106] Jianbing Ni, Xiaodong Lin, Kuan Zhang, and Yong Yu. 2016. Secure and deduplicated spatial crowdsourcing: A fog-based approach. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'16)*. 1–6.

[107] Ruifang Niu, Wenfang Song, and Yong Liu. 2013. An energy-efficient multisite offloading algorithm for mobile devices. *Int. J. Distrib. Sens. Netw.* 9, 3 (2013), 1–6.

[108] OECI. 2017. Retrieved from http://openedgecomputing.org/index.html.

[109] Opeyemi Osanaiye, Shuo Chen, Zheng Yan, Rongxing Lu, Kim Choo, and Mqhele Dlodlo. 2017. From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access* (2017).

[110] Jesus Pacheco and Salim Hariri. 2016. IoT security framework for smart cyber infrastructures. In *Proceedings of the IEEE International Workshops on Foundations and Applications of Self* Systems (FAS*W'16)*. IEEE, 242–247.

[111] Seok-Hwan Park, Osvaldo Simeone, and Shlomo Shamai. 2016. Joint cloud and edge processing for latency minimization in fog radio access networks. In *Proceedings of the IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC'16)*. IEEE, 1–5.

[112] Georgios Paschos, Ejder Bastug, Ingmar Land, Giuseppe Caire, and Mérouane Debbah. 2016. Wireless caching: Technical misconceptions and business barriers. *IEEE Commun. Mag.* 54, 8 (2016), 16–22.

[113] Mach Pavel and Becvar Zdenek. 2017. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutor.* 19, 3 (Thirdquarter 2017), 1628–1656.

[114] Mugen Peng, Shi Yan, Kecheng Zhang, and Chonggang Wang. 2016. Fog-computing-based radio access networks: Issues and challenges. *IEEE Netw.* 30, 4 (2016), 46–53.

[115] Xuhong Peng, Ju Ren, Liang She, Deyu Zhang, Jie Li, and Yaoxue Zhang. 2018. BOAT: A block-streaming app execution scheme for lightweight IoT devices. *IEEE IoT J.* 5, 3 (2018), 1816–1829.

[116] Konstantinos Poularakis, George Iosifidis, Vasilis Sourlas, and Leandros Tassiulas. 2014. Multicast-aware caching for small cell networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'14)*. IEEE, 2300–2305.

[117] Konstantinos Poularakis, George Iosifidis, and Leandros Tassiulas. 2013. Approximation caching and routing algorithms for massive mobile data delivery. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'13)*. IEEE, 3534–3539.

[118] Kuacharoen Pramote, J. Mooney Vincent, and K. Madisetti Vijay. 2003. Identity-based authentication scheme for the Internet of Things. In *Proceedings of the Conference on Design, Automation and Test in Europe Conference (DATE'03)*. IEEE, 912–917.

[119] Lingjun Pu, Xu Chen, Jingdong Xu, and Xiaoming Fu. 2016. D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE J. Sel. Areas Commun.* 34, 12 (2016), 3887–3901.

[120] Praveen Kumar Rajendran. 2015. Hybrid intrusion detection algorithm for private cloud. *Ind. J. Sci. Technol.* 8, 35 (2015).

[121] Buvaneswari A. Ramanan, Lawrence M. Drabeck, Mark Haner, Nachi Nithi, Thierry E. Klein, and Chitra Sawkar. 2013. Cacheability analysis of HTTP traffic in an operational LTE network. In *Proceedings of the IEEE Wireless Telecommunications Symposium (WTS'13)*. IEEE, 1–8.

[122] Ammar Rayes and Salam Samer. 2016. *Internet of Things From Hype to Reality: The Road to Digitization*. Springer.

[123] Ju Ren, Hui Guo, Chugui Xu, and Yaoxue Zhang. 2017. Serving at the edge: A scalable IoT architecture based on transparent computing. *IEEE Netw.* 31, 5 (2017), 96–105.

[124] Ju Ren, Yundi Guo, Deyu Zhang, Qingqing Liu, and Yaoxue Zhang. 2018. Distributed and efficient object detection in edge computing: Challenges and solutions. *IEEE Netw.* 32, 6 (2018), 137–143.

[125] Ju Ren, Yaoxue Zhang, Ruilong Deng, Ning Zhang, Deyu Zhang, and Xuemin Shen. 2016. Joint channel access and sampling rate control in energy harvesting cognitive radio sensor networks. *IEEE Trans. Emerg. Top. Comput.* 7, 1 (2019), 149–161. doi:10.1109/TETC.2016.2555806.

[126] Morabito Roberto, Cozzolino Vittorio, Ding Aaron Yi, Beijar Nicklas, and Ott Jörg. 2018. Consolidate IoT edge computing with lightweight virtualization. *IEEE Netw.* 32, 1 (2018), 102–111.

[127] Tiago Gama Rodrigues, Katsuya Suto, Hiroki Nishiyama, and Nei Kato. 2017. Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control. *IEEE Trans. Comput.* 66, 5 (2017), 810–819.

[128] Ola Salman, Sarah Abdallah, Imad H Elhajj, Ali Chehab, and Ayman Kayssi. 2016. Identity-based authentication scheme for the Internet of Things. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'16).* IEEE, 1109–1111.

[129] Mahadev Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (2017), 30–39.

[130] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Cáceres, and Nigel Davies. 2009. The case for VM-based cloudlets in mobile computing. *IEEE Perv. Comput.* 8, 4 (2009), 14–23.

[131] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. 2015. Edge analytics in the Internet of Things. *IEEE Perv. Comput.* 14, 2 (2015), 24–31.

[132] Min Sheng, Chao Xu, Junyu Liu, Jiongjiong Song, Xiao Ma, and Jiandong Li. 2016. Enhancement for content delivery with proximity communications in caching enabled wireless networks: Architecture and challenges. *IEEE Commun. Mag.* 54, 8 (2016), 70–76.

[133] Adeshina Busari Sherif, Mumtaz Shahid, Al-Rubaye Saba, and Rodriguez Jonathan. 2018. 5G millimeter-wave mobile broadband: Performance and challenges. *IEEE Commun. Mag.* 56, 6 (2018), 137–143.

[134] Peng Shu, Fangming Liu, Hai Jin, Min Chen, Feng Wen, Yupeng Qu, and Bo Li. 2013. eTime: Energy-efficient transmission between cloud and mobile devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13).* IEEE, 195–199.

[135] Javad Sohankar, Koosha Sadeghi, Ayan Banerjee, and Sandeep K. S. Gupta. 2015. E-bias: A pervasive eeg-based identification and authentication system. In *Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'15).* ACM, 165–172.

[136] Hong Song, Dacheng Wang, Jiaxin Wang, and Wenhao Zhang. 2017. TC-CCS: A cooperative caching strategy in mobile transparent computing system. In *Proceedings of the IEEE IEEE International Conference on Visual Communications and Image Processing (VCIP'17).* IEEE, 1–4.

[137] Hongguang Sun, Matthias Wildemeersch, Min Sheng, and Tony QS Quek. 2015. D2D enhanced heterogeneous cellular networks with dynamic TDD. *IEEE Trans. Wireless Commun.* 14, 8 (2015), 4204–4218.

[138] Xiang Sun and Nirwan Ansari. 2017. Latency aware workload offloading in the cloudlet network. *IEEE Commun. Lett.* 21, 7 (2017), 1481–1484.

[139] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. 2017. On multi-access edge computing: A survey of the emerging 5G network edge architecture & orchestration. *IEEE Commun. Surv. Tutor.* 19, 3 (2017), 1657–1681.

[140] Wenjuan Tang, Kuan Zhang, Ju Ren, Yaoxue Zhang, and Xuemin Sherman Shen. 2019. Flexible and efficient authenticated key agreement scheme for BANs based on physiological features. *IEEE Trans. Mobile Comput.* 18, 4 (2019), 845–856.

[141] Yayuan Tang, Kehua Guo, and Biao Tian. 2018. A block-level caching optimization method for mobile transparent computing. *Peer-to-Peer Network. Appl.* 11, 4 (2018), 711–722.

[142] Stefano Traverso, Mohamed Ahmed, Michele Garetto, Paolo Giaccone, Emilio Leonardi, and Saverio Niccolini. 2013. Temporal locality in today's content caching: Why it matters and how to model it. *ACM SIGCOMM Comput. Commun. Rev.* 43, 5 (2013), 5–12.

[143] D. Trihinas, A. Tryfonos, M. D. Dikaiakos, and G. Pallis. 2018. DevOps as a service: Pushing the boundaries of microservice adoption. *IEEE Internet Comput.* 22, 3 (2018), 65–71.

[144] Luis M. Vaquero and Luis Rodero-Merino. 2014. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Comput. Commun. Rev.* 44, 5 (2014), 27–32.

[145] Ricard Vilalta, Raluca Ciungu, Arturo Mayoral, Ramon Casellas, Ricardo Martinez, David Pubill, Jordi Serra, Raul Munoz, and Christos Verikoukis. 2016. Improving security in Internet of Things with software defined networking. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'16).* IEEE, 1–6.

[146] Todd Vollmer and Milos Manic. 2014. Cyber-physical system security with deceptive virtual hosts for industrial control networks. *IEEE Trans. Industr. Inf.* 10, 2 (2014), 1337–1347.

[147] Serdar Vural, Ning Wang, Pirabakaran Navaratnam, and Rahim Tafazolli. 2017. Caching transient data in internet content routers. *IEEE/ACM Trans. Netw.* 25, 2 (2017), 1048–1061.

[148] Huaqun Wang, Zhiwei Wang, and Josep Domingo-Ferrer. 2017. Anonymous and secure aggregation scheme in fog-based public cloud computing. *Fut. Gener. Comput. Syst.* 78, 1 (2018), 712–719.

[149] Rui Wang, Xi Peng, Jun Zhang, and Khaled B. Letaief. 2016. Mobility-aware caching for content-centric wireless networks: Modeling and methodology. *IEEE Commun. Mag.* 54, 8 (2016), 77–83.

[150] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, and Wenbo Wang. 2017. A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access* 5, 3 (2017), 6757–6779. DOI: https://doi.org/10.1109/ACCESS.2017.2685434

[151] Xiaofei Wang, Min Chen, Tarik Taleb, Adlen Ksentini, and Victor Leung. 2014. Cache in the air: Exploiting content caching and delivery techniques for 5G systems. *IEEE Commun. Mag.* 52, 2 (2014), 131–139.

[152] Xiumin Wang, Jin Wang, Xin Wang, and Xiaoming Chen. 2015. Energy and delay tradeoff for application offloading in mobile cloud computing. *IEEE Syst. J.* 11, 2 (2015), 858–867.

[153] Yanting Wang, Min Sheng, Xijun Wang, Liang Wang, and Jiandong Li. 2016. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* 64, 10 (2016), 4268–4282.

[154] Shinae Woo, Eunyoung Jeong, Shinjo Park, Jongmin Lee, Sunghwan Ihm, and KyoungSoo Park. 2013. Comparison of caching strategies in modern cellular backhaul networks. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (Mobisys'13).* ACM, 319–332.

[155] Huaming Wu, Yi Sun, and Katinka Wolter. 2018. Energy-efficient decision making for mobile cloud offloading. *IEEE Trans. Cloud Comput.* (2018). DOI: https://doi.org/10.1109/TCC.2018.2789446

[156] Changqiao Xu, Tianjiao Liu, Jianfeng Guan, Hongke Zhang, and Gabriel-Miro Muntean. 2013. CMT-QA: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks. *IEEE Trans. Mobile Comput.* 12, 11 (2013), 2193–2205.

[157] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014. A survey of information-centric networking research. *IEEE Commun. Surv. Tutor.* 16, 2 (2014), 1024–1049.

[158] Jianxiao Yang, Benoît Geller, and Stéphanie Bay. 2011. Bayesian and hybrid Cramér–Rao bounds for the carrier recovery under dynamic phase uncertain channels. *IEEE Trans. Sign. Process.* 59, 2 (2011), 667–680.

[159] Lei Yang, Jiannong Cao, Hui Cheng, and Yusheng Ji. 2015. Multi-user computation partitioning for latency sensitive mobile cloud applications. *IEEE Trans. Comput.* 64, 8 (2015), 2253–2266.

[160] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky. 2014. Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing. In *Proceedings of the IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'14).* 325–329. DOI: https://doi.org/10.1109/CAMAD.2014.7033259

[161] Shanhe Yi, Cheng Li, and Qun Li. 2015. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 ACM Workshop on Mobile Big Data (Mobidata'15).* ACM, 37–42.

[162] Changsheng You and Kaibin Huang. 2016. Multiuser resource allocation for mobile-edge computation offloading. *arXiv preprint* arXiv:1604.02519 (2016).

[163] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. 2017. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wireless Commun.* 16, 3 (2017), 1397–1411.

[164] Engin Zeydan, Ejder Bastug, Mehdi Bennis, Manhal Abdel Kader, Ilyas Alper Karatepe, Ahmet Salih Er, and Mérouane Debbah. 2016. Big data caching for networking: Moving from cloud to edge. *IEEE Commun. Mag.* 54, 9 (2016), 36–42.

[165] Guanglin Zhang, Wenqian. Zhang, Yu. Cao, Demin Li, and Lin Wang. 2018. Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices. *IEEE Trans. Industr. Inf.* 14, 10 (2018), 4642–4655.

[166] Ke Zhang, Supeng Leng, Yejun He, Sabita Maharjan, and Yan Zhang. 2018. Cooperative content caching in 5G networks with mobile edge computing. *IEEE Wireless Commun.* 25, 3 (2018), 80–87.

[167] Ke Zhang, Supeng Leng, Yejun He, Sabita Maharjan, and Yan Zhang. 2018. Mobile edge computing and networking for green and low-latency Internet of Things. *IEEE Commun. Mag.* 56, 5 (2018), 39–45.

[168] Ke Zhang, Yuming Mao, Supeng Leng, Yejun He, and Yan Zhang. 2017. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Vehic. Technol. Mag.* 12, 2 (2017), 36–44.

[169] Ke Zhang, Yuming Mao, Supeng Leng, Quanxin Zhao, Longjiang Li, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. 2016. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE Access* 4, 8 (2016), 5896–5907.

[170] Meng Zhang, Hongbin Luo, and Hongke Zhang. 2015. A survey of caching mechanisms in information-centric networking. *IEEE Commun. Surv. Tutor.* 17, 3 (2015), 1473–1499.

[171] Weiwen Zhang, Yonggang Wen, and Dapeng Oliver Wu. 2015. Collaborative task execution in mobile cloud computing under a stochastic wireless channel. *IEEE Trans. Wireless Commun.* 14, 1 (2015), 81–93.

[172] Yaoxue Zhang. 2004. Transparence computing: Concept, architecture and example. *Chin. J. Electr.* 32, S1, Article 169 (2004), 5 pages.

[173] Yaoxue Zhang, Kehua Guo, Ju Ren, Yuezhi Zhou, Jianxin Wang, and Jianer Chen. 2017. Transparent computing: A promising network computing paradigm. *Comput. Sci. Eng.* 1, 2 (2017), 7–20.

[174] Yaoxue Zhang, Ju Ren, Jiagang Liu, Chugui Xu, Hui Guo, and Yaping Liu. 2017. A survey on emerging computing paradigms for big data. *Chin. J. Electr.* 26, 1 (2017), 1–12. DOI: https://doi.org/10.1049/cje.2016.11.016

[175] Yaoxue Zhang, Laurencetianruo Yan, Yuezhi Zhou, and Wenyuan Kuang. 2013. Information security underlying transparent computing: Impacts, visions and challenges. *Web Intell. Agent Syst.* 8, 2 (2013), 203–217.

[176] Yaoxue Zhang and Yuezhi Zhou. 2006. Transparent computing: A new paradigm for pervasive computing. In *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC'06).* Springer-Verlag, Berlin, 1–11. DOI: https://doi.org/10.1007/11833529_1

[177] Yaoxue Zhang and Yuezhi Zhou. 2007. 4VP: A novel meta OS approach for streaming programs in ubiquitous computing. In *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA'07).* IEEE, 394–403.

[178] Yaoxue Zhang and Yuezhi Zhou. 2013. Transparent computing: Spatio-temporal extension on von Neumann architecture for cloud services. *Tsing. Sci. Technol.* 18, 1 (Feb. 2013), 10–21.

[179] Tianchu Zhao, Sheng Zhou, Xueying Guo, Yun Zhao, and Zhisheng Niu. 2015. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing. In *Proceedings of the IEEE GLOBECOM Workshops.* IEEE, 1–6.

[180] Yun Zhao, Sheng Zhou, Tianchu Zhao, and Zhisheng Niu. 2015. Energy-efficient task offloading for multiuser mobile cloud computing. In *Proceedings of the IEEE International Conference on Cognitive Computing (ICCC'15).* IEEE, 1–5.

[181] Zhongyuan Zhao, Mugen Peng, Zhiguo Ding, Wenbo Wang, and H Vincent Poor. 2016. Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks. *IEEE J. Select. Areas Commun.* 34, 5 (2016), 1207–1221.

[182] Cong Zuo, Jun Shao, Guiyi Wei, Mande Xie, and Min Ji. 2018. CCA-secure ABE with outsourced decryption for fog computing. *Future Gener. Comput. Syst.* 78, 1 (2018), 730–738.

Received January 2019; revised July 2019; accepted September 2019