IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 8, 2019, accepted July 1, 2019, date of publication July 9, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927538

Edge Computing and Networking: A Survey on Infrastructures and Applications

YONGLI ZHAO[1], **(Senior Member, IEEE), WEI WANG**[1], **YAJIE LI**[1],
CARLOS COLMAN MEIXNER[2], **MASSIMO TORNATORE**[3,4], **AND JIE ZHANG**[1]
[1]Institute of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1TH, U.K.
[3]Department of Electronics and Information, Politecnico di Milano, 20133 Milan, Italy
[4]Department of Computer Science, University of California at Davis, Davis, CA 95616, USA

Corresponding authors: Yongli Zhao (yonglizhao@bupt.edu.cn) and Wei Wang (weiw@bupt.edu.cn)

This work was supported in part by the National Science and Technology Major Project under Grant 2017ZX03001016 and in part by the State Key Laboratory of Advanced Optical Communication Systems Networks of China.

ABSTRACT As a concept to enhance and extend cloud-computing capabilities, edge computing aims to provide Internet-based services in the close proximity to users by placing IT infrastructures at the network edge in forms of tiny datacenters. Taking advantage of the close distance to end user and access networks, edge datacenters can provide low-latency and context-aware services and further improve users' quality of experience. As the network edge is a geographically spread concept, the edge datacenters are usually highly distributed so that they can provide nearby storage and processing capabilities to most of the end users. Furthermore, edge datacenters also co-work with centralized cloud datacenters for service orchestration. Such decentralization and collaboration are expected to introduce significant transformations to both infrastructures and applications. To provide an overview of how edge can be integrated with cloud-computing and how edge computing can benefit applications, this paper studies the infrastructure and application issues of edge computing and networking in several sub-aspects, including related concepts, infrastructures, resource management and virtualization, performance, and applications.

INDEX TERMS Edge computing, cloud-computing, datacenter, SDN, NFV, C-RAN, IoT.

I. INTRODUCTION

The success and rapid adoption of cloud computing is creating new business opportunities; meanwhile, it is also posing complex technical challenges to the Information Technology (IT) industry. It is expected that the global cloud-computing market by 2024 will reach $1 Trillion [1]. New cloud computing applications are emerging, among which several of them are requiring low-latency and high bandwidth service. Thus, the traditional cloud computing architecture, which is usually composed of only a few large data centers interconnected by long distance networks, is being challenged by these new emerging applications, as these centralized Data-Centers (DCs) are usually far away from end users and cannot provide ultra-low latency and high bandwidth connectivity. To deal with these challenges, edge computing was introduced as a supplemental paradigm for cloud computing, in which data, computing/storage capacity and applications are distributed in network edge (typically in metro segment of the network). Located at the place that is closer to users, edge computing provides an intermediate anchor between the users and cloud, and thus extends cloud-computing capabilities to the network edge.

In edge computing, computing and storage capacity are provided by the distributed "edge DCs". With edge DCs, traffic from some applications might be able to be served at network edge, and thus can avoid traveling long distances to centralized cloud DCs. Therefore, edge DCs can reduce the amount of traffic to the cloud, and decrease the transmission latency of network connectivity. In addition, edge DCs can also help avoiding links and nodes at high risk of congestion, disruption, and cyber-attack.

To achieve the above-mentioned benefits of edge computing, many challenges that are introduced by the distributed computing paradigm must be addressed. First, regarding the topic of constructing computing system, the distributed edge

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

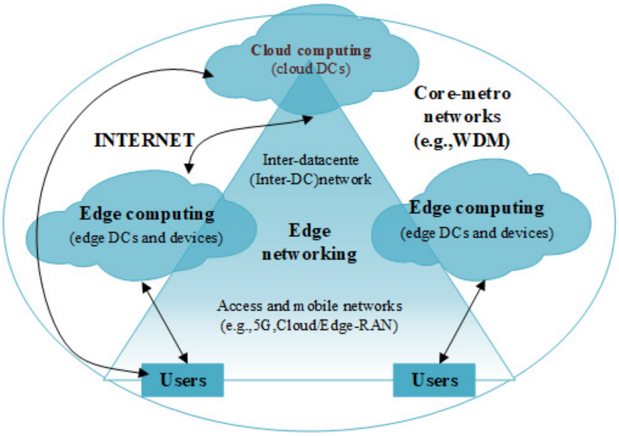

FIGURE 1. **Edge computing and networking.**

infrastructures must be integrated together to build an edge computing system. Second, over the edge infrastructures, the distributed computing/storage capacity must be orchestrated to support different applications. Fig. 1 shows a general architecture of edge computing, in which edge DCs reside between the large scale of users and the centralized cloud DCs. To provide cloud-like service to end users, edge DCs must be connected to the cloud DC and end users through network infrastructures (we will refer such edge-DC related networking issues as ''edge networking''). At the DC side, edge DCs are connected to the cloud DC through the core/metro networks; at the user side, edge DCs are connected to users through the access networks. Moreover, the distributed edge DCs with networking infrastructures between users and the cloud DC, along with the corresponding resources they can provide, must be managed and orchestrated properly for cloud-like service provisioning.

In summary, edge computing is not only a technology about computing itself, but also an architectural concept that has many impacts on the related ecosystem. First, the presence of distributed IT resources at network edge will introduce transformations to the infrastructures in traditional cloud-computing DCs. Second, the computing capability that is provided by edge DCs can enable more edge-based service paradigms. Thus, with edge computing, the Internet will be no longer a centralized service paradigm, because edge DCs will take care of some Internet traffic directly. To study how edge infrastructures support the new service paradigms and related applications, this paper studies edge computing and networking from infrastructures and applications perspective.

Few surveys have already appeared on edge computing, but their focus and contribution are different with that in our work. Ref. [2] surveyed the state-of-the-art MEC research from the communications perspective. Specifically, this survey summarized the MEC models of computation tasks, communications, mobile devices and MEC servers, and analyzed the latency and energy consumption model accordingly. Ref. [3] studied the edge resource allocation and mobility management issues in mobile networks. Specifically, it reviewed recent advances on opportunistic offloading techniques covering both traffic and computation offloading protocols and techniques working in an opportunistic context or behavior. It also compares different offloading techniques in various aspects, e.g., realization method, applicable scenario, advantages and drawbacks. Since edge computing can provide not only computing capability, but also storage at network edge, Ref [4] studied computation offloading and cloud-edge cooperation issues from a caching perspective. Specifically, it studied the research progress on content popularity, caching policies, scheduling, and mobility management. It also elaborated on the application and use cases of mobile edge networks, including cloud technology, software defined network, network function virtualization and smarter mobile devices. From the security perspective, Ref. [5] surveys security threats, challenges, and mechanisms in edge computing, and studies the synergies among security mechanisms originally designed for edge paradigms and other related fields. In summary, existing surveys have covered many issues that are introduced by the edge-computing paradigm. Applications in edge computing have been mentioned, but most of them are the specific use cases in the context they focused on, e.g., radio optimization, offloading, caching. Instead of focusing on the issues that are introduced by edge computing, this paper will discuss the infrastructures issue that supports edge computing, and study how related infrastructures can construct the edge computing system and enable various applications.

The rest of the paper is organized as follows. Section II introduces several similar concepts (i.e., cloud, grid, edge, mobile, and hybrid computing) in the context of edge computing. Section III discusses the infrastructure of edge computing, and resource management issues, including the control plane and resource virtualization. Section IV reviews some existing works on the performance issues of edge computing (i.e., latency, energy saving, security, resiliency etc.). Section V discusses the applications of edge computing in several use cases.

II. RELATED CONCEPTS

Besides cloud and edge computing, there are some similar definitions about other computing paradigms. To clarify the connections and differences between edge computing and other similar computing paradigms, this section reviews several computing concepts, including grid computing, cloud computing, edge computing, mobile cloud computing, fog computing, and multi-access edge computing.

A. GRID COMPUTING

Grid computing was proposed in the early 1990s as a special type of parallel computing that relies on multiple computers that are connected to a network to perform a large-scale sharing of processing or storage resources. With grid computing, underlying distributed computing infrastructures can be collaboratively utilized to support cross-organizational applications [6]. Grid computing is often used for the problems that involves extensive numerical calculation, which can be easily parallelized. One numerical calculation task can be divided into multiple sub-tasks, and each machine on a grid might be assigned with a sub-task. For each sub-task, after being processed, its result will be sent back to the primary machine, which takes care of all sub-tasks. After receiving the results of all the sub-task, the primary machine will execute further processing on the received interim results to generate final output. Therefore, grid computing focuses on multiple-party cooperation to achieve a final goal effectively.

B. CLOUD COMPUTING

Cloud computing was first introduced in 1996 as a network-based computing paradigm. However, the first use of ''cloud computing'' in its modern context occurred in 2006, and it was used to describe the new paradigm in which people access software, computing capabilities, and files over the Web instead of their local desktops [7]. In other words, it provides storage and processing capabilities in an on-demand manner by relying on high-capacity DCs that are accessible through Internet, and this setup allows users to access their data and applications from anywhere as long as they are connected to Internet [8]. Since then, cloud computing has been widely adopted as a famous computing paradigm.

There are four types of cloud computing models available in the market: public cloud, private cloud, hybrid cloud, and community cloud [9]. From the technical aspect, there is no much significant difference for implementing these four types of cloud computing. However, their customers, scales and security levels are different. Public cloud provide computing service through public internet, and its users are usually individuals or small companies. For enterprises, which need higher computing capabilities, they usually build their own private cloud. Hybrid cloud is a mixture of public and private cloud (e.g., VMware cloud on AWS). Similarly, a community cloud falls between public and private cloud, as some organizations setup a separate private cloud for themselves, called a community cloud.

Different with parallel calculation tasks in grid computing, applications in cloud computing do not access the required resources directly; rather it accesses them through something like a service. So instead of talking about a specific hard drive for storage, and a specific CPU for computing, cloud computing talks the service that provides these resources. Such service maps any requests for resources to its physical infrastructures. Usually the service has access to a large amount of physical resources, and can dynamically allocate the resources on demand.

C. EDGE COMPUTING

Edge computing appeared around 2002, and it refers to the paradigm that allows computation resources to be located at the network edge, so that computing occurs near data sources [10]. An edge device can be any kind of computing or networking infrastructure residing between data sources and cloud DCs. For example, it could be a smartphone sitting between body sensors and cloud DCs, or a micro-DC/cloudlet between a mobile device and cloud DC.

Edge devices play as a kind of middleware between user device and cloud DCs. Basically, edge devices can be used in two different fashions. First, they can take over some tasks, which originally should be handled by user's device locally, and this case is referred to as task offloading. Another case is that edge devices can take parts of cloud DC's jobs, and this case is actually kind of extension of cloud computing [11].

D. MOBILE CLOUD COMPUTING

Mobile cloud computing (MCC) emerged as an extension of cloud computing in mobile networks. MCC aims at offloading the processing tasks and the storage tasks of mobile devices to computational powerhouses. Since 2009, it has become one of the industry buzzwords and a major discussion thread in the IT world [12]. MCC integrates the advantages of mobile computing, mobile internet and cloud computing. In MCC, there are four types of cloud-based resources: distant immobile clouds, proximate immobile computing entities, proximate mobile computing entities, and hybrid (combination of the other three model) entities.

MCC is a network-scenario-specific name, and thus cannot be compared with the above three concepts in parallel. MCC is a concept with obvious heterogeneity [13], and the mobile cloud here can be a general remote loud, a cloud that is composed of end user devices, or a cloudlet. The unique feature of MCC is that it is a computing paradigm in mobile networks, and thus many mobile communications problems are discussed in its context. Besides, MCC inherits the advantages of cloud computing, which can not only enhance processing, energy, and storage capabilities of mobile devices, but also amend data safety and security, data ubiquity, accessibility, and user interface.

E. FOG COMPUTING

Fog computing was first proposed in 2012, and its idea is to extend cloud-computing services towards the network edge [14]. Fog computing is an extension of cloud computing to meet the requirements of emerging applications in Internet of Things (IoT). Several different definitions of fog computing have been proposed. First, there is a comprehensive definition of fog computing [15], i.e., ''Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous), ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties''. Second, fog computing is also defined as ''a geographically distributed computing architecture with a resource pool, which consists of one or more ubiquitously-connected heterogeneous devices at the network edge, where these devices are not exclusively and seamlessly backed by cloud services'' [16]. From these definitions, we see that the goal of fog computing is to provide elastic computing, storage and networking resources collaboratively to a large scale of clients from network edge.

TABLE 1. Comparison of edge computing related concepts.

Computing	Birth	IT Infrastructure	Placement Style	Location	Low Latency	Bandwidth Capacity	Major Motivation
Grid	Early 1990s	Computers	Distributed	Anywhere	N/A	N/A	Collaborative Computation
Cloud	$\approx$1996	DC	Centralized	Core	No	Low	Internet service
Mobile Cloud	2009	DC	Centralized	Mobile Core	Yes	Medium	Mobile Internet
Edge	$\approx$2002	Micro-DC	Distributed	User - Cloud	Yes	High	Task offloading
Fog	2012	Micro-DC	Distributed	Access Net	Yes	High	IoT Application
Multi-access Edge	2014	MEC Servers	Distributed	Base Station	Yes	High	Ultra-low latency

Fog computing is application-specific name, and its unique feature is that its client devices are massive "things". Thus, in fog computing, many IoT specific problems (e.g., massive connectivity, etc.) are discussed. However, technically, fog computing has many overlaps with edge computing, and it shares some of the technical problems with it.

F. MULTI-ACCESS EDGE COMPUTING

Multi-access Edge Computing (MEC) [17], which was originally defined as Mobile edge computing, aims to provide IT and cloud-computing capabilities within the Radio Access Network (RAN). For application developers and content providers, RAN edge offers ultra-low latency and direct access to real-time radio network information (such as subscriber location, cell load, etc.) that can be used by applications to support context-related services; these services are capable of differentiating mobile broadband experiences. With MEC, content, services and applications can be accelerated from network edge. Mobile subscriber's experience can also be enriched through efficient operations, based on the insights into the radio and network conditions.

It is notable that MEC is primarily motivated by latency and quality of experience, and thus MEC is a performance-motivated name. Its unique features has two major aspects. One is to provide computing resources at a closer proximity to users, and thus can reduce network latency. The other one is to take advantage of the real-time network channel information for further optimization. MEC also has overlaps with other concepts, especially with edge computing.

G. DISCUSSION

This section presents six similar computing concepts, and Table. 1 compares them in several dimensions, including the architecture, location of IT infrastructures, latency and bandwidth capacity. Note that all parameters are compared in a relative manner. It is notable that the former three, grid computing, cloud computing and edge computing, are defined in a general fashion, and can be distinguished easily. The latter three, edge computing, fog computing and MEC, are similar, but they target on different scenarios/applications. Specifically, the location of MEC servers is generally closer to end users. In particular, MCC has overlaps with both cloud computing and edge computing. Fig. 2 further shows the relationship of these similar concepts, and we can see that the common idea of edge computing, fog computing and MEC is that they have IT and networking infrastructures at the close-user proximity.

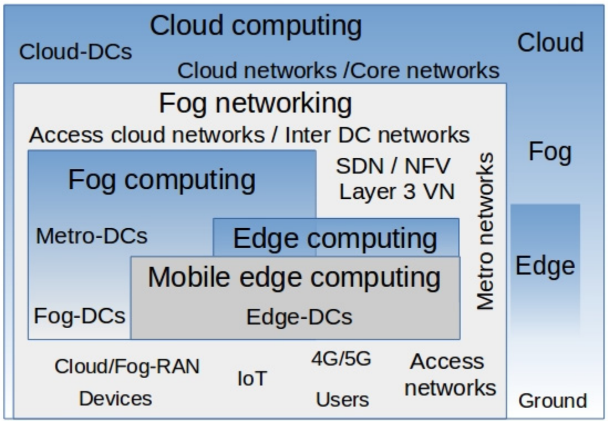

FIGURE 2. Cloud vs. edge computing and networking.

In the following parts, we will NOT focus on the difference of the edge-related concepts; instead, we will take the term "edge computing" as a representative of them to discuss their common idea of providing IT capacities at the network edge.

III. INFRASTRUCTURE AND RESOURCE MANAGEMENT FOR EDGE COMPUTING AND NETWORKING

To provide computing and storage capabilities at the close proximity to end users, edge computing brings new IT and network infrastructures to the network edge. First, edge computing is no longer a centralized computing paradigm; instead, the edge DCs that host computing and storage resource are distributed at the network edge. Second, the networking infrastructure around edge DCs need to provide connectivity for edge DCs to communicate with user devices and cloud DCs. This section discusses the infrastructure and resource issues for edge computing in three aspects, which are: i) infrastructure; ii) control and management; iii) resource virtualization and orchestration; and 4) system architecture.

A. INFRASTRUCTURES

We define the infrastructure for edge computing as a set of distributed components, and categorize them in two domains: i) for computing, physical IT components that support heterogeneous and distributed processing and storage, ii) for networking, physical network components that support communication between distributed computing elements.

1) IT INFRASTRUCTURE

In the sense that edge computing is a kind of extension of cloud computing, cloud IT infrastructures are still parts of the edge computing system. In addition, edge computing also has its IT infrastructures that are distributed network edge. In particular, the edge side infrastructures are heterogeneous in different derivations of edge computing. Considering these infrastructures together, this section form the IT infrastructure in edge computing as three types, which are: i) cloud DC; ii) edge side DC, and iii) user side device.

a: CLOUD SIDE

Cloud DC: provides large amount of centralized devices or servers for data processing and content placement in the core of the network. There are usually tens of thousands, even hundreds of thousands of servers in a single cloud DC, and they can serve millions of users in a client/server manner. Cloud DC in edge does not has much difference with that in cloud computing, and we will not comment much on it.

b: EDGE SIDE

Edge DC: is smaller than a cloud DC (i.e., between tenths to hundreds servers), and it is usually placed close to the end users devices, typically in the metro network or in the edge of metro network, e.g., network providers' central offices. In the derivations concepts of edge computing, edge DC might be named as other terms, as follows.

Cloudlet: In the context of MCC, the IT infrastructure is called Cloudlet, and it refers to a local computing node that comprised of several multi-core computers with connectivity to the remote cloud servers [18]. With Cloudlet, mobile devices' workloads can be offloaded locally.

Fog DCs/Servers: In fog computing, the IT infrastructure is usually called fog DC/servers, which are geo-distributed and are deployed at very common places for example; bus terminals, shopping centers, roads, parks, etc. [19].

MEC Server: Edge DC is also called MEC servers in the context of MEC. MEC servers at the edge of the RAN usually have some insights to the real-time radio and network information (such as subscriber location, cell load, etc.) that can be leveraged by applications and services to offer context-related services. Such services are capable of differentiating the mobile broadband experience [20].

c: USER SIDE

User Side Device: Compared with edge DCs, they are nearer to users, and can be part of a router, a gateway of IoT, a personal computer, a laptop, or a smart phone, in which there are processing, memory, and storage resources. In some cases, the user side devices are formed as a micro/nono DC to work in a collaborative manner [21]. Note that these devices are usually from individual users, and their availability depend on users' permission for sharing their personal computing and storage resources with others.

2) NETWORK INFRASTRUCTURE

The other part of infrastructure in edge computing is the network components, which provide connectivity between the cloud DCs, edge DCs, and users' devices. We category them into two layers: i) access networks, which connect user devices with edge DCs; and ii) transport networks that connect the edge DCs with cloud DCs.

a: ACCESS NETWORKS

According to the specific location of edge DCs, end users could be connected to edge DCs through one of the three different access technologies, i.e., cellular wireless networks, passive optical networks (PON), and fiber-wireless access networks,. In the case where edge DCs are placed at the wireless access point, users can access edge DCs directly through the radio channels. In this case, cooperative communications ability of wireless channels can help to optimize task offloading in the form of relaying via the nearer mobile device [22]. In C-RAN, the fronthaul connects the radio head and baseband processing unit (BBU), and PON is a promising technology for fronthaul transport. When edge DC is deployed in the central office that hosts the BBU, users' traffic is aggregated to the edge DC through the cellular channels and PON [23]. In addition, in the same case of edge computing in C-RAN, the wireless access party might be the typical RAN technologies (i.e., WLAN) instead of the cellular channels [24]. In this case, the connectivity between users and edge DC will be the typical RAN and PON, which is also referred to as the fiber-wireless network.

b: TRANSPORT NETWORKS

In general, transport networks do not have much difference with that in cloud computing, as it is not aware of the traffic flows for specific services. However, from the end-to-end view, transport networks, including the packet and optical domains, need to be orchestrated with cloud/edge DCs and user devices to provide connectivity for edge-cloud resource sharing and synchronization [25].

B. CONTROL AND MANAGEMENT

Edge computing introduces distributed edge DCs in infrastructure layer, but the distributed edge DCs still need to serve users in a cloud-like fashion, which means users should not be aware of such distribution. Therefore, the distribution of edge DCs and related infrastructures must be managed properly to provide service in a united manger, and this requirement poses significant challenges to the control and management plane. To understand how to integrate the edge DCs with existing cloud computing infrastructures, this section reviews the existing works on the control and management issues of edge computing.

1) CENTRALIZED CONTROL

The distribution of IT and network infrastructures in edge computing somehow goes the opposite directions with the united service manner of cloud computing. Fortunately, SDN, the recently emerging control technique, fits such requirement of edge computing very well, and it can provide centralized management and virtualization abilities over distributed devices [26]. According to the role of SDN and edge computing in an integrated system, SDN works with edge computing in the following fashions.

a: SDN-CONTROLLED EDGE

The centralized control nature of SDN fits the distributed feature of edge computing very well, and thus SDN can be enhanced to control edge elements [27]. From the system perspective, such enhancement is implemented by adopting the control logic and protocol to edge features. Here are some examples. Yiming and et al. developed a prototype of edge computing control plane based on SDN, and modified the SDN controller to enable edge DC control functionalities. In particular, such integrated controller can serve as a platform for performing analytics by collecting data from edge DCs [28]. Targeting on the IoT specific use cases, Ibrahim and et al. implemented a SDN-based control platform for edge computing to orchestrate the caching in Information Centric Networks [29]. In the context of Internet of Vehicles (IoV), SDN was enhanced with the ability to manage various heterogeneous entities and features (e.g., physical devices, mobility and capability) in Vehicular Adhoc Networks [30].

b: EDGE-POWERED SDN

As we know, SDN controllers are implemented as software instances that run on IT infrastructures. Edge computing extends the IT infrastructures and capabilities, and thus offers more options for deploying SDN controllers. Such case is not that popular, but it can benefit SDN in some special cases. For example, in vehicle communications networks, the edge-based SDN architecture can distribute the control components to the edge infrastructures and further to reduce the response time of control messages. To be more specific, the control plane in such architecture can acquire the position, direction, velocity, and network connectivity in real time [31].

c: SDN CO-WORK WITH EDGE

Edge computing works based on distributed IT elements, and SDN is born for control distributed network elements. Based on such connection, they can work together to achieve some common objectives. For example, In IoT, SDN can manages the IoT traffic, and the edge-based gateway is able to secure the IoT traffic. Such cooperation enables a model for securing IoT devices using SDN and edge computing [32].

2) MANAGEMENT PROTOCOLS

In edge computing, management protocols are used to connect the control plane with distributed edge entities, including the cloud/edge DCs and the network infrastructures. In different application contexts of edge computing, there are different management protocols and they affect the performance of edge computing systems. Among the existing management protocols, three popular network management protocols, i.e., Constrained Application Protocol (CoAP), Simple Network Management Protocol (SNMP), and NETCONF, were evaluated in the general context of edge computing [33], and their performance was compared in three cases, where the management traffic is routed in different fashions. The cases are, the management plane are connected with end devices: i) directly; ii) through local gateway; and iii) through cloud servers. As expected, the latency of management traffic via cloud is the longest, and it suggest that edge computing can also benefit the control protocol by providing local connectives.

3) RESOURCE MANAGEMENT AND ORCHESTRATION

Besides controlling the distributed infrastructures in a centralized manner, the computing and networking resources in distributed infrastructures shall also be orchestrated for cloud-like service provisioning. This section discusses the resource orchestration issue in three perspectives, i.e., load balance, multi-site collaboration and Device-to-Device (D2D) communications/computing.

a: LOAD BALANCING

In edge computing, which provides multiple edge DCs as the candidate service points, users' workload can be dispatched to different edge DCs to achieve load balancing. Since user's workload varies as time going on, the load distribution among different edge DCs may need to be adjusted accordingly. According the strategies and objectives used for balancing, users' traffic can be dispatched in different ways. For example, the workload of each edge DC can be assigned based on the dynamic graph partitioning approach, and such dynamic load balancing can effectively reduce intra-DC migration [34]. In mobile networks, the variation of users' load is more significant, as users' location is changing as well. In this case, there is a network-sharing model, called Honeybee, for balancing the load of independent jobs among heterogeneous mobile nodes [35]. In general, the dynamic computations of most load balancing strategies are usually complex when the load condition changes frequently. In this aspect, the complexity of load balancing algorithms for edge computing can be optimized by clustering edge DCs into small-cells [36].

b: MULTI-SITE COLLABORATION

Edge DCs are typically close to users, and can provide low latency and high QoE services. However, the edge-DC capacity is usually limited, to manage the infrastructure cost. Thus,

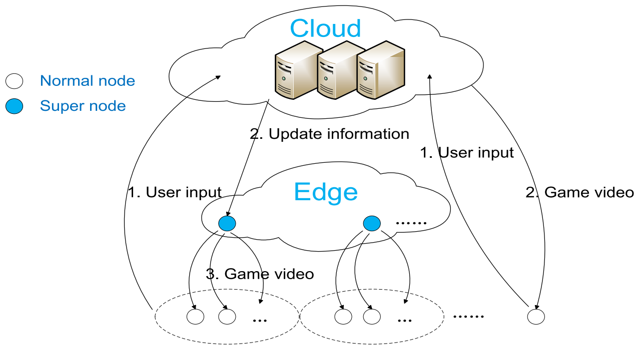

FIGURE 3. **Fog-assisted cloud gaming infrastructure.**

one individual edge DC is not able to work independently, and it may need cloud's support when necessary. Collaboration in edge computing has two major categories: horizontal and vertical collaboration [37], which refer to the collaboration among the IT elements within one layer (e.g., among user devices or edge DCs), and the collaboration across user devices, edge DCs and clouds layers, respectively.

Horizontal collaboration, which coordinates the same layer elements, can be used to achieve load balancing (as discussed above) and failure recovery. In case of overload or failure of edge DC, which will significantly degrade the QoE and negate the advantages of edge computing, horizontal collaboration plays an important role for service recovery [21]. One recovery scheme focuses on the collaboration in edge DC layer. Once overload or failure happens to an edge DC, the users, who are originally served by it, will try to offload its workload to another available edge DC within transfer range. Focusing on the collaboration among user devices, the other recovery scheme is designed for the situations when there is no available neighboring edge DCs within transfer range. In this case, some nearby user devices will be chosen as ad-hoc relay nodes, and be used to bridge the affected user devices with unreachable edge DCs.

Vertical collaboration is an approach that can coordinate user devices, edge and cloud DCs for performance optimization. In the system perspective, the existing cloud-computing software stacks can be extend to edge DCs to enable seamless service collaboration and orchestration [38]. By using the so-called External Drivers in this solution, any organization can develop their own drivers to support new equipment in edge computing system. With the system level support for vertical collaboration, application level vertical collaboration can also be achieved. CloudFog is an online game example, which aims to achieve high user QoE through vertical collaboration [39]. As shown in Fig. 3, CloudFog is constructed by a set of super-nodes that are responsible for rendering game videos and streaming them to nearby players. In this case, cloud only help edge to handle intensive game state computation and send update information to super-nodes, and this paradigm can reduce game latency and bandwidth consumption significantly.

c: D2D COMMUNICATION/COMPUTING

Taking advantage of the close distance between user devices and edge devices, end user devices are able to offload part of their tasks to the nearby devices via ultra-low latency connectivity. Such collaboration is also called D2D communication/computing, and it is well studied in the wireless-oriented edge computing (or in other words, the MCC). In edge computing, the device in D2D communications could be a user device or an edge DC. By offloading tasks through D2D communications, some devices can benefit in terms of battery standby time, local computation density, etc., paying off additional inter device traffic and extra workload to the target device. The collaborative D2D communication/computing raises several optimization issue in the context of resource orchestration, to decide when and where to offload task executions.

From the infrastructure perspective, two contrasting optimization objectives for D2D communications/computing emerge: 1) the radio efficiency of cellular networks (which provide connectivity to the devices) and 2) the efficiency of devices themselves. Focusing on workload of cellular networks introduced by D2D communications, Ref. [40] has surveyed the concept of ''opportunistic offloading'', which is based on the idea of green cellular networks. Combining the advantages of latency tolerance of some application, collaborative caching and computing resource sharing, opportunistic offloading is deemed as a promising solution for traffic orchestration and task offloading. Some other works aimed at optimizing the computing part for the system. For example, taking the communication capacity as a constraint, a recent work [41] has formulated a mixed integer non-linear model to improve the number of supported devices in a given system, and to improve the overall computing capacity further.

C. INFRASTRUCTURE RESOURCE VIRTUALIZATION

With above mentioned infrastructures and control with management techniques, service providers are able to provide edge services. However, the service provisioning based on the physical infrastructures are usually not flexible and scalable enough, as the bare-metal infrastructure vendors cannot cover all the heterogeneous demands of end users. Thanks to the virtualization technology, we are able provision IT and network resources more flexibly through abstraction, aggregation, partition, and migration of sliced resources. This subsection focuses on the virtualization technique for edge computing and networking infrastructures.

1) IT INFRASTRUCTURE VIRTUALIZATION

IT virtualization is relatively mature, and this section will not comment much on itself; instead, we review the existing works that are related with edge computing.

Virtual Machine (VM) is the most popular form for IT virtualization. With VM-based virtualization, infrastructure hypervisors can slice a physical computing node into one or more VMs, and each VM can be easily used and managed to perform isolated computing tasks. Container is another emerging IT virtualization technique. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably when it is migrated from one computing environment to another. This section discusses how VM and container can benefit edge computing.

a: VIRTUAL MACHINE

VM is an operating-system level virtualization technique. By creating a scalable system of multiple independent virtual computing devices, idle computing resources can be allocated and used more efficiently. With no doubt, VM is playing an important role in edge computing, like what it acts in cloud computing. Mahadev et al., has made a proof of concept on VM-based cloudlets in mobile computing and suggested that VM can indeed help to customize the infrastructure for diverse applications [17]. In VM-based edge computing, VM is the representative of IT resources, and they can be placed, scheduled, or event migrated to meet the heterogeneous requirements of different applications.

VM Placement/Scheduling: In VM-based edge computing, VM is the entity that provides application services. To deploy an edge computing service over distributed infrastructures, corresponding VMs need to be placed/scheduled appropriately to meet service's performance demands. To achieve the optimal VM placement/scheduling, several parameters must be taken into consideration to manage cost or improve service performance.

First, service latency is a very sensitive performance metric in this topic, as VM placement/scheduling decides the geographical location and amount of service capacity. To be more specific, the location of service entity affects the network distance and thus communication latency between client and server. In addition, the service capacity that the placed VM can provide sets the service speed, and thus affects the application layer latency [42]. Based on the effects that VM placement/scheduling has on latency, VM scheduling can be formulated as a an offline VM planning model to optimize VM placement/scheduling in edge computing, while meeting the given latency budget.

Second, energy consumption is another metric that is related with VM placement/scheduling, as the energy consumption of the infrastructure that hosts VMs corresponds to its utilization. If the VMs can be placed/scheduled appropriately to use the free resource of some powered on infrastructure, the energy efficiency can be improved. For example, based on the graph theory, one application can be placed as multi-components (in terms of VMs) to optimize energy consumption [43].

Besides, cost is another abstracted metric related with VM placement/scheduling, and it exists in almost every component in an edge computing system. In this context, a comprehensive cost model of edge computing is proposed, including: (i) the communication cost between users and servers; (ii) the execution cost at each server; and (iii) the relocation cost. Based on this model, VMs can be placed based on the sample-average-approximation to minimize the overall cost [44]. Further, the network cost and latency can be taken into account together in VM placement.

In addition, there are some demonstrations regarding VM placement in edge computing. For example, to support VM placement and resource assignment, a SDN-based control framework is designed for edge computing [45]. In the context of service chaining, a simulation platform, called Network CloudSim, is developed to demonstrate the orchestration and consolidation of service chains placement in edge computing [46].

VM Migration: Besides providing the flexible manner for placing/ scheduling services over physical IT infrastructures, VM also enables another fancy way to manage IT resources – VM migration, which is impossible without virtualization. In edge computing, VMs that act as the entity for task offloading or network-based application servers also can be migrated sometimes to achieve performance optimization.

First, at the edge of mobile networks, mobility is a significant challenge for both network itself and the service VMs. The user experience may be degraded when user moves, as the optimal condition of access networks and the previous placed VMs may be invalid. To keep track of the user mobility, the server side VMs can be migrated accordingly to achieve dynamic and continuous optimization [47].

In addition, in case of malicious attacks or system failures that edge infrastructure may experience, VMs may also need to be migrated to other sites in a proactive manner to avoid data loss and to decrease the service downtime [48].

b: CONTAINER

Container is a recent emerging application level virtualization technique, and it is deemed as a lightweight IT resource entity compared with VM [49]. Besides the benefits of VM in terms of size and flexibility, containers are specifically relevant for platform concerns that are typically dealt with Platform-as-a-Service (PaaS) clouds such as application packaging and orchestration. Existing review suggests that container has the potential to advance PaaS technology towards distributed heterogeneous clouds (including edge computing) substantially through lightweightness and interoperability [50].

Docker is a representative product of container. Its application is evaluated in edge computing in terms of i) deployment and termination, ii) resource and service management, iii) fault tolerance, and iv) caching. Previous works suggest that Docker can provide fast deployment, small footprint and good performance, and it is a potential viable platform for edge computing [51]. Moreover, it is demonstrated that edge computing can be built based on the open-source Kura gateway and docker-based containerization over edge nodes, and the demonstration shows the flexibility and easy deployment trait of docker for edge computing [52].

2) NETWORK INFRASTRUCTURE VIRTUALIZATION

Similar to IT virtualization, network virtualization is another important aspect for edge computing infrastructure virtualization, as flexible network resource provisioning will also benefit edge networking a lot. The traditional essence of network virtualization is to abstract the network infrastructure and provide virtual network connectivity with the abstracted resource. However, from the view that traditional network functions are tightly coupled with the dedicated hardware, network virtualization is also emerging in another aspect, which is Network Function Virtualization (NFV).

a: *VIRTUAL NETWORKS*

Network is a general concept that has two distinct branches, i.e., computer networks and telecom networks. Virtualization of computer network is relatively mature [53], and there are already commercial products (e.g., Cisco ACI and VMware NSX) in the market. In telecom networks, signal channels in each physical link are usually multiplexed in some dimensions, and can be manipulated individually. On such basis, various types of virtual networks can be built on the shared infrastructure for multi-tenants. The emerging of edge computing and corresponding edge infrastructures have no significant impact on telecom network virtualization itself, but have raised a new use-case that the geographical distributed infrastructures need secure connectivity to build cross-site interconnections. In this case, overlay virtual networks that are constructed over heterogeneous network infrastructures become a vital option for providing secure communication channels and protecting data privacy [54].

b: *NETWORK FUNCTION VIRTUALIZATION*

As mentioned, traditional network functions are coupled with the specific hardware and this fashion is not flexible enough to manage networking in an on-demand manner. To consolidate heterogeneous network equipment into industry standard high volume servers, switches and storage, NFV is emerging as another aspect of network virtualization for provisioning virtualized network functions over IT infrastructures [55]. NFV can help to facilitate the networking for edge computing, and edge computing, which providing IT infrastructures at network edge, in turn can help to build NFV.

NFV For Edge: Together with SDN, NFV can enhance the interoperability of edge nodes. For example, powered by NFV/SDN, edge nodes can be orchestrated and integrated with cloud DCs to achieve E2E orchestration [56].

EDGE for NFV With edge computing in computer networks, some gateways can be deployed as virtual machines in a local edge cloud, and this is the case where SDN-enhanced cloud at network edge plays as a cornerstone for NFV.

c: *NETWORK SLICING*

Network slicing is a higher-level abstraction of networks, and it is kind of composition of virtual networks with NFV. With the intension of accommodating service with heterogeneous QoS, network slicing is naturally compatible with the distributed paradigm of edge computing. To meet user-customized demand on QoS, network slicing can be coordinated with edge computing allocation for integrated optimizations. Here are some examples. In beyond 5G, the wireless resource can be sliced according to the transmission power of the base-station, where the traffic is offloaded to edge servers [57]. In IoT, transport networks can be sliced together with IoT service resources [58]. Taking energy as the optimization objective, edge computing resources can also be sliced (distribute workload among edge nodes) dynamically to provide energy-efficient partitions/slices of computing resources [59].

D. SYSTEM ARCHITECTURES

This section discusses the architectures for edge computing and networking in two perspectives: i) infrastructure architecture, ii) service framework.

1) INFRASTRUCTURE ARCHITECTURE FOR EDGE COMPUTING

As discussed, the distributed edge DCs usually need to work collaboratively, and the corresponding resources shall be orchestrated as a system for united service provisioning. Fig. 4 illustrated an example infrastructure architecture of edge computing [60]. In this architecture, edge DCs provide low-latency and high-bandwidth services to the IoT devices and moving vehicles nearby through the Edge Service Gateway (ESG), and all the edge DCs are back connected to the central DC. The central DCs act as the backup for edge DCs, and they can also handle the traffic, which has been pre-processed by the edge or do not even require edge processing.

2) SERVICE FRAMEWORK FOR AN EDGE SYSTEM

The physical edge computing infrastructures can provide computing, storage and networking resources, but they are not capable of serving users' traffic directly, unless corresponding service-oriented functional and operational service entities are equipped. Fig. 5 shows an example framework from the perspective of functionality components, and it has three layers, i.e., mobile device layer, cloudlet layer, and cloud layer. In mobile device layer, there is a sub-layer edge application-programming interface, which provides a set of pre-defined functions, such as task offloading, application monitoring, cloudlet discovery and synchronization, and application migration. Cloudlet layer offers edge-based entities and services, i.e., VMs (or containers) that can host applications and data. Cloud layer focuses on orchestration functions, such as replication and load balancing.

Discussion: As discussed above, edge computing brought significant transformation to the infrastructure layer. Fig. 6 shows an overview of the connections between edge and other related components. First, the access and transport networks provide physical connectivity for edge DCs to communicate with users and cloud. Second, similar with cloud DCs, edge DCs can host applications directly, or host

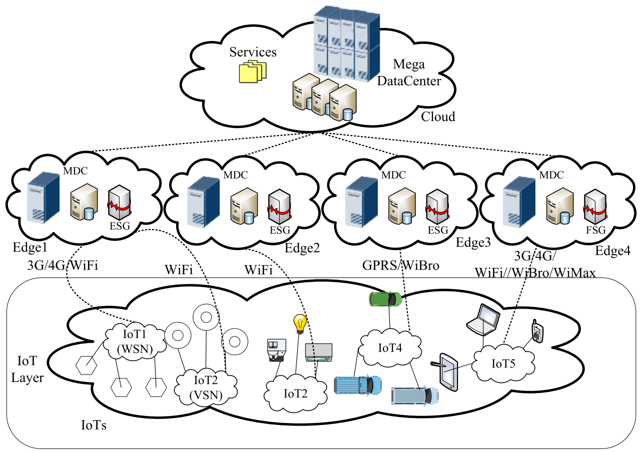

FIGURE 4. Edge computing architecture in support to IoT.

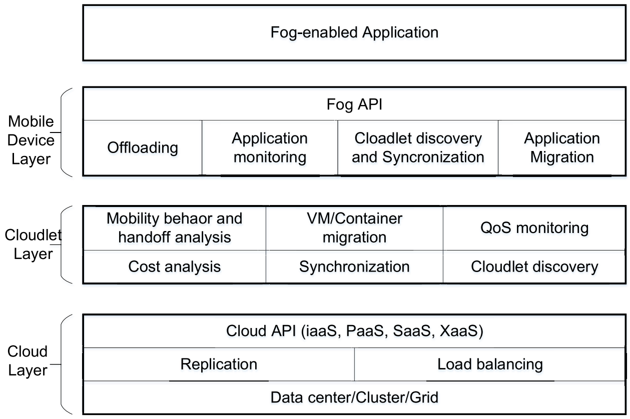

FIGURE 5. Layered architecture of edge computing.

VMs/Containers to provide network-based services. Third, edge DCs can host NFV instances to facilitate network virtualization, and the NFV instances may serve edge DCs in turn. Fourth, network slicing can be built based on the VMs/Containers and NFV instances at edge DCs. Fifth, edge DC might be managed by SDN controller, and it can also host SDN controllers. At last, the virtualized entities in edge DC can be orchestrated with that in peer edge DCs or in cloud DCs to achieve load balancing, multi-site collaboration, failure recovery, etc. Based on the infrastructure, the following sections will discuss the performance issues and applications.

3) AVAILABLE PLATFORMS FOR EDGE COMPUTING

Currently, available platforms for edge computing are mainly divided into two categories, i.e., open source frameworks and commercial products. Linux Foundation launches some projects for edge computing in IoT, such as EdgeX Foundry and Akraino Edge Stack. Some Internet companies like Baidu and Huawei also announce their open source edge computing platform such as OpenEdge and Kubeedge. These frameworks are usually software with more than a dozen micro services that provide an interface to the IoT and serve as an edge layer to connect the client and the cloud. Commercial products include AWS by Amazon, Azure by Microsoft and Edge TPU by Google. AWS and Azure promote cloud service business to the edge, ensuring low latency for services. Edge TPU is the ASIC designed to run AI with high performance and low power at the edge.

IV. SERVICE PROVISIONING AND PERFORMANCE OPTIMIZATION

Born for mission-critical applications, edge computing is expected to be capable of providing high-quality services to uses. To meet the performance requirements of the mission-critical applications, several performance issues need to be addressed. This section focuses on the issues of latency, energy consumption, security and resiliency.

A. LATENCY AND ENERGY-CONSUMPTION

Distributed at the close-proximity to users, edge DCs can generally reduce the service latency by avoiding long-distance network transmission. However, latency is not an independent metric in edge computing; instead, it is coupled with energy consumption, as more active computing and networking nodes can principally serve more users and process users' traffic faster. Thus, this section will discuss latency and energy consumption together.

1) LATENCY

As edge DCs are geographically distributed, the location of edge DCs sets the network transmission latency, which is caused by the distance for fetching corresponding services in edge DCs. In addition, given a certain edge DCs, the workload assigned to each edge DC decides the application layer processing latency. Thus, service latency is primarily determined by service placement and workload assignment.

In the context of service scheduling and placement, latency violations must be considered when making service-scheduling decisions for latency-critical services. It is true that deploying more VMs can reduce the latency violation, but they need to consume more infrastructure resource and further affect operators' cost and revenue. Regarding this tradeoff, a Lyapunov optimization framework for edge computing is developed to schedule VMs to meet the service SLAs, while maximizing the revenue of infrastructure operator [61].

Focusing on the impact of workload allocation on service latency in edge-cloud computing, a workload allocation problem was formulated to assign workloads to edge and cloud DCs towards the minimal power consumption with the service latency as the major constraint [62]. The problem was then tackled using an approximate approach by decomposing the primal problem into three sub-problems of corresponding subsystems, which can be respectively solved. Simulations results show that by sacrificing modest computation resources to save communication bandwidth and reduce transmission latency, edge computing can significantly improve the performance of cloud computing.

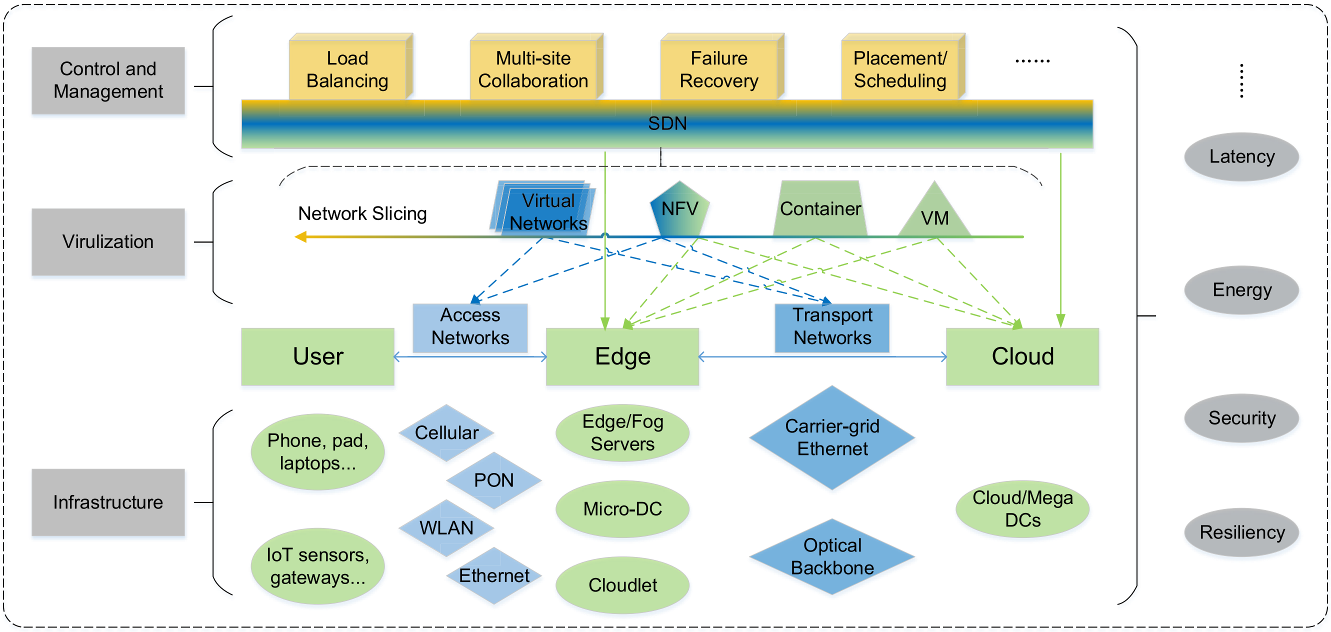

FIGURE 6. Overview of the connections between edge and other infrastructures.

In addition, taking both VM placement and workload assignment into account, an ILP model was developed to minimize the hardware consumption for running VMs, under the latency constraint [42].

2) ENERGY-EFFICIENCY

From the infrastructure perspective, the energy consumption in edge computing and cloud computing is mainly caused by edge/cloud DCs, network equipment and user devices.

As to the edge/cloud DCs, it is a common sense that the energy-efficiency of cloud DCs is higher than that of distributed edge DCs. However, cloud DCs are far away from users and assigning workloads to them will incur higher network latency. To coordinate edge and cloud DCs properly, an energy-aware interplay between edge and cloud was investigated considering SDN-based WAN network [63], which proves that it is possible to handle edge traffic in a latency-aware manner by decreasing average path length without significant deterioration in terms of blocking probability and carbon footprint.

As to the user devices, there are two critical but conflicting objectives, i.e., the power consumption and the execution latency of computation tasks in user devices, and they are determined by the task-offloading policies. There is a formulation and an online algorithm for task offloading, and it decides local execution and computation offloading to minimize power consumption through dynamic service migration, while taking the task-buffer stability as the major constraints [64]. In the context of wireless cellular networks, the task scheduling is formulated as a constrained stochastic shortest path problem on a directed acyclic graph to minimize the energy consumption of user device [65]. This problem is addressed under three alternative wireless channel models, including: i) a block-fading channel, ii) an IID stochastic channel, and iii) a Markovian stochastic channel.

In addition, edge computing was used to build an energy management platform for residential, industrial, and commercial areas [66]. The scalability, adaptability, and open source software/hardware in this platform enable the user to implement energy management functions with the customized control-as-services, while minimizing the implementation cost and time-to-market.

B. SECURITY

Security is a general problem for information system, and it applies to edge computing as well. On the one hand, the distinctive characteristics (location sensitivity, wireless connectivity, geographical accessibility, etc.) of edge computing may introduce new security issues and challenges. On the other hand, the computing capability provided by edge computing at network edge might be able to help to improve the security level of a certain applications.

In edge computing, the authentication, access control, intrusion detection, and privacy are major security issues [67]. More specific, the unique security threats introduced by IoT in edge computing consist of man-in-the-middle attack, intrusion detection, malicious codes, and malicious users [68]. There is an application, called Selective Encryption and Component-Oriented Deduplication (SEACOD) for guarding the security of edge computing, and it can achieve both fast and effective data encryption and reduction for edge computing services [69]. Specifically, SEACOD efficiently de-duplicates redundant objects in files, emails, as well as images exploiting object-level components based on their

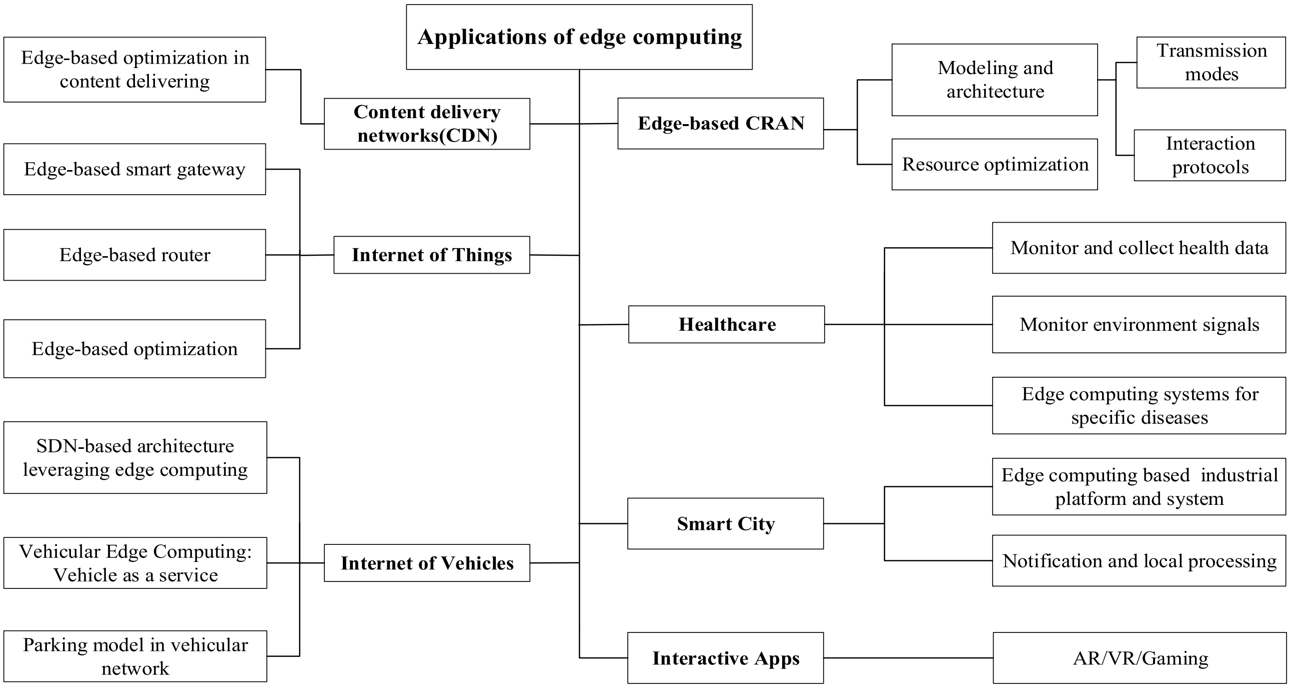

FIGURE 7. A taxonomy of edge computing applications.

structures. It also effectively reduces the overall encryption overhead on the mobile devices by adaptively applying compression and encryption methods according to the decomposed data types.

From the perspective of how to improve security using edge computing, several edge-based security approaches were studied at the application and infrastructure layer. In application layer, two edge computing-based approaches were applied to prevent the data thief attack by deploying decoy information within the edge DCs and within personal online social networking profiles [70]. In infrastructure layer, there is a policy-based resource management plane for supporting secure collaboration and interoperation between different user-requested resources in edge computing [71].

C. RESILIENCY

Resiliency denotes the recover ability of a system in case of failure, and it is a key issue in edge computing as that in cloud computing. Resiliency strategies in cloud computing have been summarized by a survey [72]. In edge computing, some DCs (i.e., edge DCs, and micro-DCs) are located at network edge, but the resiliency issues of the DC themselves still applies. Hence, we will also refer readers to this survey. Since the network for edge computing is not an edge-specific system, most research on network resiliency can be applied to the network part in edge computing, and please refer to [73] for more details about network resiliency.

V. APPLICATIONS OF EDGE COMPUTING

Edge computing can reduce the service latency and decrease bandwidth demand in core networks. With these advantages, edge computing can serve many latency-critical and bandwidth-intensive applications, including C-RAN, CDN, IoT/IoV, healthcare, smart city and interactive applications, as is shown in Fig. 7. This section discusses the connection between edge computing and these applications.

A. CLOUD RAN IN 5G

C-RAN was proposed by China Mobile [74] in 2010, and it has become a hot topic in 5G mobile communications area. C-RAN aims to virtualize the mobile network Base Band processing Unit (BBU) using standard IT servers in central offices. In other words, C-RAN also needs computing and storage resources at network edge. Thus, the idea of C-RAN is compatible with edge computing, and C-RAN is a very significant use case of edge computing.

First, edge-computing resources can be used to serve users' traffic at network edge directly, and thus reduce end-to-end latency. For example, edge computing can help mobile applications on task offloading and mobile storage expansion. Second, edge-computing resources can also be used to support virtualized C-RAN functions. Since survey work [75] has summarized the details of edge computing for C-RAN from various points of view, including contents caching, mobility management, and radio access control, we will not comment much here.

In addition to the concept of C-RAN, there are some other similar use cases. Fog-based RANs (F-RAN) was presented in [76], and it architecture is shown in Fig. 8. Compared with the traditional RANs (globe centralized mode), it is notable that F-RAN (local centralized mode) has the radio signal processing and radio resource management

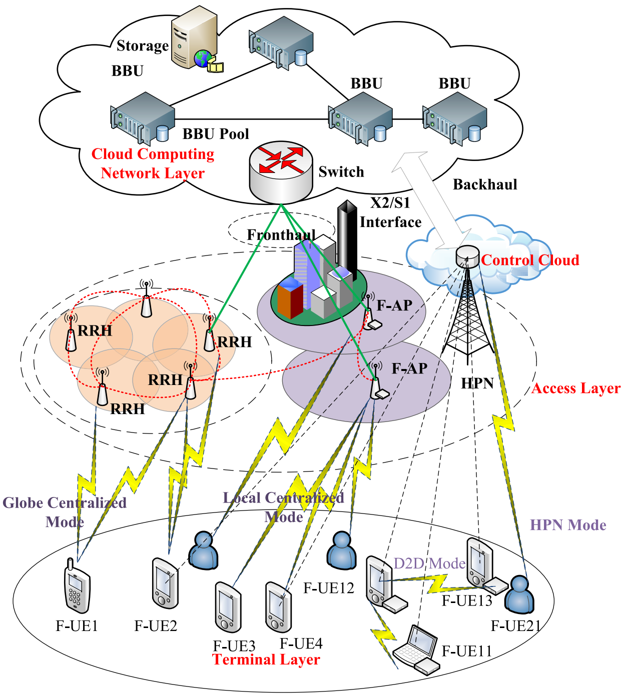

FIGURE 8. System model for implementing F-RANs.

unit (F-AP) implemented at network edge, where is closer to end users.

From the perspective of application implementation, there is a programming model and framework for developing applications based on edge computing. In this framework, application developers can leverage a common application mindset to design elastic and scalable edge-based mobile applications [77]. It is shown that the application developers can benefit from this model by accessing the features of edge computing easily. Another prototype also showed that edge computing could provide an effective and sometimes better alternative to help implementing mobile applications [78].

B. CONTENT DELIVERY NETWORK

Content delivery network (CDN) is a globally distributed network overlaid on the current network infrastructure and it is composed of multiple datacenters that provide content to end-users with high availability and high performance service. CDN can serve a large range of content delivering applications, such as web objects, live streaming media, and social networks. Edge DCs, which have storage resources distributed network edge, are natural entities for content caching. In addition to caching contents simply, edge DCs can fully leverage the computing resources to further process the local cached content and deliver the most desirable services to mobile users [79]. Beyond CDN, edge computing can also optimize web-based applications using its knowledge about networks' context and users' conditions (e.g. network status and device's computing load) [80].

C. IoT

IoT is featured by large amount of end devices at network edge, and most of the end devices are lightweight sensors, which usually do not have sufficient resources for complex processing. In this case, IoT devices usually need to offload their data to computing-rich entities through the Internet [81]. By providing infrastructures for such offloading, edge computing is widely used to implement IoT-specific devices. Further, edge computing can also be used to implement the control plane for managing IoT devices.

From the perspective of implementing IoT-specific devices, edge computing can be used to implement IoT gateways and edge routers. IoT gateway is the first hop entity for aggregating traffic from distributed IoT devices. Generally, a smart IoT gateway can be built with edge computing to preprocess and trim data in edge DCs before sending it to the cloud [82]. In the context of wireless IoT, edge computing can enable the IoT gateway for wireless sensors and actuators networks [83]. This gateway consists of master nodes and slave nodes, and can manages virtual gateway functions, flows, and resources. Beyond IoT gateway, edge computing can also be used to build the adaptive edge router, which has the ability on regressive admission control and fuzzy weighted queueing. This edge computing-based router can monitor and react to network QoS changes within heterogeneous networks, and can optimize the network performance and resources accordingly. In addition, a novel framework for a new generation of IoT devices was studied based on edge computing, and it enable multiple new features for both the IoT administrators and end users on the basis of edge computing [84]. Exploiting the recent emergence of SDN, such smart IoT devices can also be used to build fast, reliable and diverse IoT applications.

From the perspective of managing IoT-related devices, edge computing can help in the efficient, effective, and fair management of resources for the IoT and other underlying devices. SOFT-IoT is an example platform for facilitating the local computing processing and more complex operations running on virtual entities in IoT [85]. In SOFT-IoT, part of data processing capacity and service delivery operations are processed locally in ''small servers'' (edge DCs), which are close to where the data is collected. To deal with the issues of resource prediction, customer-type-based resource estimation and reservation, advance reservation, and pricing for new and existing IoT customers, edge computing is used in a dynamic resource-estimation and pricing model [86].

D. IoV

IoV is a special use-case of IoT, but IoV has dedicated demands on mobility and low latency. This section discusses how edge computing can be applied in IoV from the architecture and service perspectives.

From the architecture perspective, there is an architecture, called Vehicular Adhoc Networks (VANETs), for IoV, and it is built with SDN and edge computing [87]. As shown

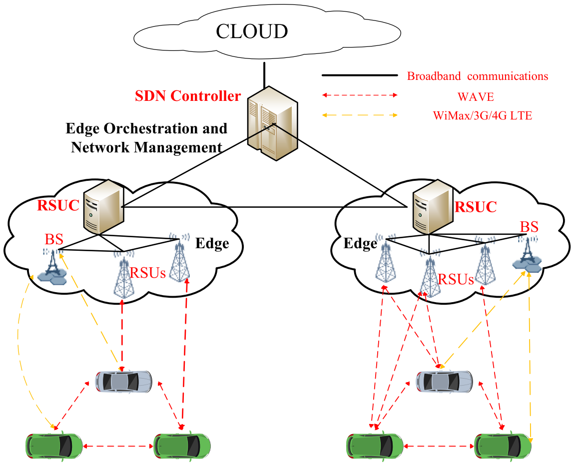

FIGURE 9. SDN-based VANET architecture leveraging edge computing.

in Fig. 9, the VANET architecture is composed of two parts: i) SDN-based control plane, providing flexibility, programmability and global knowledge; ii) roadside units, exploiting mobility prediction to decide which data they should fetch from the Internet and to schedule transmissions to vehicles. Besides assisting IoV with edge computing, there is another architecture, called Vehicular Fog Computing (VFC), in which vehicles are used as the edge infrastructures [30]. Based on the utilization of individual communication and computing resources in each vehicle, VFC uses a multitude of end-user clients and near-user edge devices to support collaborative communication and computation.

From the service perspective, edge computing offers delay-sensitive and location-aware services to IoV. For example, vehicle traffic may experience congestions when the number of vehicles increase rapidly, and finding a parking space is remarkably difficult and expensive [88]. In this case, edge computing is used in a shared parking model in to assist parking using match theory in edge and cloud.

E. HEALTHCARE

With the advantages of ultra-low latency, edge computing is applied to healthcare to support real-time operations. Based on the role of edge computing, most edge-computing-based healthcare applications are designed for patient/body monitoring. That is because edge computing can speed up the data processing for these applications.

With edge computing, the real-time bio-signals from sensor nodes can be sent to the gateway, and further transmitted to edge DCs for deep processing, visualization, and diagnosis. There are several healthcare monitoring solutions based on edge computing, including the portable fall detection system, i.e., U-Fall [89], and the Chronic Obstructive Pulmonary Disease and Mild Dementia monitoring system [90]. In addition, edge computing can also provide advanced techniques and services, such as embedded data mining, distributed storage, and notification service to healthcare. For example, an augmented brain computer interface based on edge computing was reported to detect users' brain states in real-life situations from wireless headsets, smart phones and ubiquitous computing services [91].

In addition to monitoring the body signals, edge computing can also be used to monitor environment signals that are related with healthcare. First, edge computing can work with wearable devices to collect clinical speech data from the patients with Parkinson's disease [92]. Second, mobile phone cameras can be used to detect ultraviolet, and the mobile side results are gathered and amended to edge DCs to achieve more accurate ultraviolet measurement [93].

F. SMART CITY

In a geographical perspective, the network edge usually resides with cities, as most end users live there. Therefore, edge computing has a variety of use cases in smart city.

In cities, emergency events notification is a case that has strict demands for latency, and edge computing can optimize the latency by offering computing resource at the close proximity. To provide a fast channel for notifying the relevant emergency dealing department, a smart phone based service, i.e., Emergency Help Alert Mobile Cloud is built based on edge computing [94]. In this case, edge DCs gather the location of incident and automatically contacts the emergency dealing department. The emergency related information is then synchronized automatically from edge to cloud, allowing further analysis and improvement in safety of the people. Similarly, based on the cooperation among the base transceiver stations (edge DCs), edge computing is used in another solution to rapidly notify the users, who are close to the critical area [95].

Besides the urgent cases that need low latency and quick response, edge computing can also help to build platforms for citywide data processing. For example, edge computing is used to: i) build the intelligent performance evaluation system for food cold chain [96]; ii) perform series prediction over the set of data to estimate the temperature at a future point in time; and 3) support smart living and dataflow analysis [97].

G. INTERACTIVE APPLICATIONS

Interactive applications are expected to be the killer applications for edge computing. As of now, there are three popular interactive applications: Augmented Reality (AR), Virtual Reality (VR) and online gaming. Compared with regular applications, interactive applications typically have more strict requirements on service latency. Thus, besides acting as the closest service points to provide the required context and data to interactive applications with low latency, edge DCs, with the help of context-specific auxiliary devices and optimization strategies, are also used to facilitate interactive features.

For AR in mobile networks, edge DCs equipped with GPUs take the computation-intensive traffic that is offloaded from clients to accelerate computer-vision tasks, and thus can provide desired end-to-end latency to mobile AR [98].

TABLE 2. Role of edge and special requirements for different applications.

	C-RAN	CDN	IoT	IoV	Healthcare	Smart City	Interactive Apps
Role of Edge	NFV	Storage/ Caching	NFV/Data&Task offloading	Data&Task offloading	Data&Task offloading	Data&Task offloading	Data&Task offloading
Extreme-low Latency	No	No	No	Yes	No	No	Yes
Location Tracking	No	No	Depends	Yes	No	Depends	Yes
Mobility Handover	Yes	No	No	Yes	No	No	No

In VR-based gaming, edge computing, together with proactive caching and mmWave communication, are used to improve the QoE. In this case, game players, equipped with mmWave head-mounted displays, are connected to mmWave Access Points (APs), which are further connected to multiple edge servers through the edge networks. Based on the players' movement, pose and impulse actions, HD frames are generated at the edge server side and distributed to game players with imperceptible latency [99].

Discussion: As discussed above, edge computing can be used to support different kind of applications. Edge computing plays different roles for different applications, and different applications in turn have different requirements for edge computing. Note that, since we are discussing at a general application-type level, instead of referring to specific use-cases, the roles and requirements in Table 2 apply to most use-cases within each application type, but it may not cover all use-cases. As shown in Table 2, for most application types, including IoT/IoV, Healthcare, Smart City and Interactive Applications, the most common role of edge is to act as the entity to collect data from user devices and to perform corresponding processing accordingly (Data/Task Offloading). Besides, edge sites can also be used to facilitate or virtualize some network functions (NFV). For example, edge sites can support the baseband signal processing and radio resource management functions in C-RAN, and act as the IoT gateways/routers in IoT. Edge sites also can act as the entity for caching (Storage/Caching) in CDN.

Based on their unique features, different application types may enforce special requirements on edge. First, most of the edge-based applications need edge to provide low-latency services, but some applications need extreme-low latency. For example, extreme-low latency is required to provide real-time instructions in IoV to guard the safety of fast-moving vehicles, and it is also required in Interactive Apps to provide real-time context for simulated reality. In IoV and Interactive applications, since user devices are usually moving, location tracking is required to support the navigation and real/virtual context mapping. In some specific use-cases of IoT and Smart city, location tracking may also be required if the client device is a moving entity. For some applications, whose users move in a wide area, mobility handover became a problem. For example, in C-RAN, baseband signal processing function needs handover between different central offices when cellular users move to another area; similarly, in IoV, the location-aware services also need handover to a nearby edge site when a vehicle leaves one site. In summary, edge computing can facilitate many applications by acting as various roles, but it also need to deal with various requirements of different applications to meet the QoE.

VI. CONCLUSION

Edge computing extends cloud-computing capability to the network edge by providing computing, storage and networking resources at the close proximity of Internet users. To provide an overview of how edge can be integrated with cloud computing and how edge computing can benefit applications, we reviewed related works and discussed on edge computing and networking in the following four aspects. I) Related concepts, including cloud computing, grid computing, mobile cloud computing, edge computing and mobile edge computing. II) Infrastructure and resource management, including physical infrastructure, virtual infrastructure and system architecture of edge computing. III) Performance issues for edge computing, including latency, energy consumption, security, etc. IV) Applications of edge computing, including C-RAN, IoT, healthcare and smart city. In general, the occurrence of edge computing infrastructures raised new challenges for existing IT and networking systems. In this case, the IT and network virtualization techniques are very helpful for flexible and adaptive scheduling and orchestration, and we believe that there will be more and more efficient scheduling and orchestration approaches for different use-cases. Nevertheless, edge computing offers many opportunities for service provisioning and network optimization. Based on the fact that the computing and storage capacities at network edge can serve some latency and bandwidth sensitive tasks directly, we also believe that edge computing will boost more and more amazing applications for end users and network itself.

REFERENCES

[1] *Global Cloud Computing Market Forecast 2019-2024*, Tabular Anal., Jan. 2018. [Online]. Available: https://www.marketresearchmedia.com/?p=839

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

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

[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[6] B. Jacob, M. Brown, K. Fukui, and N. Trivedi, "Introduction to grid computing," IBM, Armonk, NY, USA, Tech. Rep., Dec. 2005. [Online]. Available: http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf

[7] A. Regalado, "Who coined 'cloud computing?'" MIT, Cambridge, MA, USA, Tech. Rep., Oct. 2011. [Online]. Available: https://www.technologyreview.com/s/425970/who-coined-cloud-computing/

[8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[9] S. Goyal, "Public vs private vs hybrid vs community—Cloud computing: A critical review," *Int. J. Comput. Netw. Inf. Secur.*, vol. 63, pp. 20–29, Feb. 2014.

[10] (Dec. 2002). *Server Computing and the Network Edge*. [Online]. Available: http://www.serverwatch.com

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[12] H. Qi and A. Gani, "Research on mobile cloud computing: Review, trend and perspectives," in *Proc. 2nd Int. Conf. Digit. Inf. Commun. Technol. Appl. (DICTAP)*, May 2014, pp. 195–202.

[13] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generat. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

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

[15] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[16] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[17] Y. Hu, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, ETSI White Paper 11, 2015.

[18] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, to be published.

[19] R. Mahmud and R. Buyya, "Fog computing—A taxonomy, survey and future directions," in *Internet of Everything. Internet of Things* (Technology, Communications and Computing), B. Di Martino, K. C. Li, L. Yang, and A. Esposito, Eds. Singapore: Springer, 2016.

[20] *Mobile Edge Computing (MEC); Technical Requirements*, ETSI, Sophia Antipolis, France, 2016.

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

[22] X. Hu, K.-K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.

[23] W. Wang, Y. Zhao, M. Tornatore, H. Li, J. Zhang, and B. Mukherjee, "Coordinating multi-access edge computing with mobile fronthaul for optimizing 5G end-to-end latency," in *Proc. Opt. Fiber Commun. Conf. Expo. (OFC)*, San Diego, CA, USA, Mar. 2018, pp. 1–3.

[24] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5G era," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 192–200, Feb. 2017.

[25] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez,, T. Tsuritani, and I. Morita, "Integration of IoT, transport SDN, and edge/cloud computing for dynamic distribution of IoT analytics and efficient use of network resources," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1420–1428, Apr. 1, 2018.

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

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

[28] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery," in *Proc. COMSNETS*, Jan. 2016, pp. 1–6.

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

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

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

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

[33] M. Slabicki and K. Grochla, "Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2016, pp. 1315–1319.

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

[35] N. Fernando, S. W. Loke, and W. Rahayu, "Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 329–343, Apr./Jun. 2016.

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

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

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

[39] Y. Lin and H. Shen, "Cloud fog: Towards high quality of experience in cloud gaming," in *Proc. Int. Conf. Parallel Process.*, Sep. 2015, pp. 500–509.

[40] D. Xu, Y. Li, X. Chen, J. Li, P. Hui, S. Chen, and J. Crowcroft, "A survey of opportunistic offloading," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2198–2236, 3rd Quart., 2018.

[41] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, Feb. 2019.

[42] W. Wang, Y. Zhao, M. Tornatore, A. Gupta, J. Zhang, and B. Mukherjee, "Virtual machine placement and workload assignment for mobile edge computing," in *Proc. IEEE CloudNet*, Sep. 2017, pp. 1–6.

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

[44] H. Badri, T. Bahreini, D. Grosu, and K. Yang, "Multi-stage stochastic programming for service placement in edge computing systems: Poster," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, Aug. 2017, p. 28.

[45] Y. Zhao, Y. Li, W. Wang, X. Yu, J. Zhang, H. Zheng, Y. Lin, M. Tornatore, and B. Mukherjee, "Experimental demonstration of VM designation in hybrid cloud-fog computing with software-defined optical networking," in *Proc. Asia Commun. Photon. Conf.*, 2016, pp. 1–5.

[46] M. Seufert, B. K. Kwam, F. Wamser, and P. Tran-Gia, "Edgenetwork-cloudsim: Placement of service chains in edge clouds using network-cloudsim," in *Proc. IEEE Conf. NetSoft*, Jul. 2017, pp. 1–6.

[47] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in *Proc. 3PGCIC*, Nov. 2015, pp. 1–8.

[48] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[49] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan. 2018.

[50] C. Pahl and B. Lee, "Containers and clusters for edge cloud architectures—A technology review," in *Proc. Int. Conf. Open Big Data (FiCloud)*, Aug. 2015, pp. 379–386.

[51] B. I. Ismail, E. M. Goortani, M. B. A. Karim, W. M. Tat, S. Setapa, J. Y. Luke, and O. H. Hoe, "Evaluation of docker as edge computing platform," in *Proc. ICOS*, Aug. 2015, pp. 130–135.

[52] P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over RaspberryPi," in *Proc. 18th Int. Conf. Distrib. Comput. Netw.*, 2017, p. 16.

[53] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: A survey," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 24–31, Nov. 2013.

[54] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Cross-site virtual network in cloud and fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 46–53, Apr. 2017.

[55] *Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action*, ETSI, Sophia Antipolis, France, Oct. 2012.

[56] C. Verikoukis, A. Mayoral, D. Pubill, R. Casellas, R. Martínez, J. Serra, R. Vilalta, and R. Muñoz, "End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node," in *Proc. Opt. Fiber Commun. Conf. Exhib. (OFC)*, Mar. 2016, pp. 1–3.

[57] T. Sanguanpuak, "Network slicing with mobile edge computing for micro-operator networks in beyond 5G," 2018, *arXiv:1811.01744*. [Online]. Available: https://arxiv.org/abs/1811.01744

[58] S. Husain, A. Kunz, A. Prasad, K. Samdanis, and J. Song, "Mobile edge computing with network resource slicing for Internet-of-Things," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 1–6.

[59] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2640–2654, Sep. 2018.

[60] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE PerCom Workshops*, Mar. 2015, pp. 105–110.

[61] K. Katsalis, T. G. Papaioannou, N. Nikaein, and L. Tassiulas, "SLA-driven VM scheduling in mobile edge computing," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, San Francisco, CA, USA, Jun. 2016, pp. 750–757.

[62] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, May 2016.

[63] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk, "Energy-aware fog and cloud interplay supported by wide area software defined networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[64] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Proc. IFIP Netw. Conf.*, Toulouse, France, May 2015, pp. 1–9.

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

[66] M. A. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 161–169, Aug. 2016.

[67] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. Int. Comput. Softw. Appl. Conf.*, Jul. 2015, pp. 53–59.

[68] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported Internet of Things environment," in *Proc. Int. Conf. Netw. Future (NOF)*, Sep. 2015, pp. 1–3.

[69] S. Song, B.-Y. Choi, and D. Kim, "Selective encryption and component-oriented deduplication for mobile cloud data computing," in *Proc. Int. Conf. Comput., Netw. Commun.*, Feb. 2016, pp. 1–5.

[70] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proc. IEEE CS Secur. Privacy Workshops (SPW)*, May 2012, pp. 125–128.

[71] C. Dsouza, G. J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2014, pp. 16–23.

[72] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2244–2281, 3rd Quart., 2016.

[73] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, 2010.

[74] *C-RAN International Workshop*, China Mobile Res. Inst., Beijing, China, Apr. 2010.

[75] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[76] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, "Architecture harmonization between cloud radio access networks and fog networks," *IEEE Access*, vol. 3, pp. 3019–3034, 2015.

[77] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *Proc. 12th Annu. IEEE Int. Conf. Sens., Commun., Netw. Workshops*, Jun. 2015, pp. 1–6.

[78] G. Orsini, D. Bade, and W. Lamersdorf, "Computing at the mobile edge: Designing elastic Android applications for computation offloading," in *Proc. 8th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Oct. 2015, pp. 112–119.

[79] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," 2015, *arXiv:1502.01815*. [Online]. Available: https://arxiv.org/abs/1502.01815

[80] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving Web sites performance using edge servers in fog computing architecture," in *Proc. IEEE SOSE*, Mar. 2013, pp. 320–323.

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

[82] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2014, pp. 464–470.

[83] W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim, "A gateway based fog computing architecture for wireless sensors and actuator networks," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2016, pp. 210–213.

[84] S. Misra and S. Sarkar, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, 2016.

[85] C. Prazeres and M. Serrano, "SOFT-IoT: Self-organizing fog of things," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Crans-Montana, Switzerland, Mar. 2016, pp. 803–808.

[86] K. Suto, H. Nishiyama, N. Kato, and C.-W. Huang, "An energy-efficient and delay-aware wireless computing system for industrial wireless sensor networks," *IEEE Access*, vol. 3, pp. 1026–1035, 2015.

[87] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1–6.

[88] O. T. T. Kim, N. D. Tri, V. Nguyen, N. H. Tran, and C. S. Hong, "A shared parking model in vehicular network using fog and cloud environment," in *Proc. 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Busan, South Korea, Aug. 2015, pp. 321–326.

[89] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. CIT/IUCC/DASC/PICOM*, Liverpool, U.K., Oct. 2015, pp. 356–363.

[90] O. Fratu, C. Pena, R. Craciunescu, and S. Halunga, "Fog computing system for monitoring Mild Dementia and COPD patients—Romanian case study," in *Proc. 12th Int. Conf. Telecommun. Mod. Satell., Cable Broadcast. Services (TELSIKS)*, Niš, Serbia, Oct. 2015, pp. 123–128.

[91] J. K. Zao, T. T. Gan, C. K. You, S. J. R. Méndez, C. E. Chung, Y. T. Wang, T. Mullen, and T. P. Jung, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. Int. Conf. Intell. Environ. (IE)*, Jun. 2014, pp. 374–377.

[92] A. Monteiro, H. Dubey, L. Mahler, Q. Yang, and K. Mankodiya, "Fit: A fog computing device for speech tele-treatments," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–3.

[93] B. Mei, R. Li, W. Cheng, J. Yu, and X. Cheng, "Ultraviolet radiation measurement via smart devices," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 934–944, Jun. 2017.

[94] M. Aazam and E. N. Huh, "E-HAMC: Leveraging fog computing for emergency alert service," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 518–523.

[95] M. Sapienza, E. Guardo, M. Cavallo, G. La Torre, G. Leombruno, and O. Tomarchio, "Solving critical events through mobile edge computing: An approach for smart cities," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–5.

[96] R. Y. Chen, "Fog computing-based intelligent inference performance evaluation system integrated Internet of Thing in food cold chain," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2015, pp. 879–886.

[97] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, "EHOPES: Data-centered fog platform for smart living," in *Proc. 25th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2015, pp. 308–313.
[98] W. Zhang, B. Han, and P. Hui, "Jaguar: Low latency mobile augmented reality with flexible tracking," in *Proc. ACM Multimedia Conf.*, 2018, pp. 829–831.
[99] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

YONGLI ZHAO (SM'15) received the B.S. degree in communication engineering and the Ph.D. degree in electromagnetic field and microwave technology from the Beijing University of Posts and Telecommunications (BUPT), where he is currently a Professor with the Institute of Information Photonics and Optical Communications. From 2016 to 2017, he was a Visiting Associate Professor with the University of California at Davis (UC Davis). Since 2018, he has been a Full Professor with BUPT. He has published more than 300 international journal and conference papers. His research interests include software-defined optical networks, elastic optical networks, machine learning in optical networks, and optical network security.

WEI WANG received the B.S. degree in communication engineering from the Beijing University of Posts and Telecommunications (BUPT), in 2013, where he is currently pursuing the Ph.D. degree in information and communications engineering. He was a Visiting Research Scholar with the University of California at Davis, from 2016 to 2017. His research interests include datacenter networking, network function virtualization (NFV), and multi-access edge computing (MEC).

YAJIE LI was a Guest Doctoral Student with the Optical Networks Laboratory (ONLab), KTH Royal Institute of Technology, Sweden. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Information Photonics and Optical Communication, Beijing University of Posts and Telecommunications, China. His research interests include elastic optical networks and software-defined optical networks, with a special focus on optical transport solutions for 5G networks.

CARLOS COLMAN MEIXNER received the Ph.D. degree in computer science from the University of California at Davis, Davis, CA, USA, in 2016. From 2016 to 2017, he was a Postdoctoral Researcher with the I3S Research Laboratory, University of Nice Sophia Antipolis, France. He is currently a Senior Researcher with the Department of Electrical and Electronic Engineering, University of Bristol, U.K. His research interests include disaster resilient networks, disaster resilient cloud-computing, disaster resilience optical networks, and artificial intelligence.

MASSIMO TORNATORE received the Laurea (M.Sc.) degree in telecommunications engineering from the Politecnico di Milano, Italy, in 2001 and the Ph.D. degree in information engineering, in 2006. From 2007 to 2009, he was a Post doctoral Researcher with the Computer Science Department, University of California at Davis, where he is currently an Adjunct Associate Professor. He is also an Associate Professor with the Department of Electronics and Information, Politecnico di Milano. He has coauthored about 190 conference and journal papers. His research interests include design, energy efficiency, and survivability in optical networks and group communication security.

JIE ZHANG received the bachelor's degree in communication engineering and the Ph.D. degree in electromagnetic field and microwave technology from the Beijing University of Posts and Telecommunications (BUPT), China. He is currently a Professor and the Vice Dean of the Information Photonics and Optical Communications Institute, BUPT. He has published more than 300 technical papers, authored eight books, and submitted 17 ITU-T recommendation contributions and six IETF drafts. He holds 17 patents. His research interests include architecture, protocols, and standards of optical transport networks. He has served as TPC Member for a number of conferences, such as ACP, OECC, PS, ONDM, COIN, and ChinaCom.

• • •