Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges

Mithun Mukherjee, *Member, IEEE*, Lei Shu, *Senior Member, IEEE*, and Di Wang

***Abstract*—Fog computing is an emerging paradigm that extends computation, communication, and storage facilities toward the edge of a network. Compared to traditional cloud computing, fog computing can support delay-sensitive service requests from end-users (EUs) with reduced energy consumption and low traffic congestion. Basically, fog networks are viewed as offloading to core computation and storage. Fog nodes in fog computing decide to either process the services using its available resource or send to the cloud server. Thus, fog computing helps to achieve efficient resource utilization and higher performance regarding the delay, bandwidth, and energy consumption. This survey starts by providing an overview and fundamental of fog computing architecture. Furthermore, service and resource allocation approaches are summarized to address several critical issues such as latency, and bandwidth, and energy consumption in fog computing. Afterward, compared to other surveys, this paper provides an extensive overview of state-of-the-art network applications and major research aspects to design these networks. In addition, this paper highlights ongoing research effort, open challenges, and research trends in fog computing.**

***Index Terms*—Fog computing, fog networks, fog-radio access networks, edge computing.**

I. Introduction

WITH the advancement in computing and wireless technologies, the world has witnessed a growing number of connected devices to the Internet at an unpretending rate. International Data Corporation (IDC) expected that sensor-enabled objects connected to network will rise to 30 billion by 2020 and the number of connected devices will increase ranging from 50 billion to 1 trillion consisting of 500 million sensors in U.S. factories, 212 billion available sensors, 110 million connected cars with 5.5 billion sensors, 1.2 million connected homes with 200 million sensors [1]. It is also estimated that there will be 237.1 million wearable devices by 2020 [2]. As expected, worldwide Internet of Things (IoT) market will grow up to $1.7 trillion in 2020 [3]. Recently, cloud computing [4]–[6] becomes an attractive choice while providing a cost-effective solution to massive data storage and processing.

Manuscript received June 2, 2017; revised November 25, 2017 and January 27, 2018; accepted March 4, 2018. Date of publication March 12, 2018; date of current version August 21, 2018. *(Corresponding author: Lei Shu.)*

M. Mukherjee is with the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China (e-mail: m.mukherjee@ieee.org).

L. Shu is with the School of Engineering, Nanjing Agricultural University, Nanjing 210095, China, and also with the School of Engineering, University of Lincoln, Lincoln LN6 7TS, U.K. (e-mail: lei.shu@ieee.org).

D. Wang is with the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China, and also with the College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China.

Digital Object Identifier 10.1109/COMST.2018.2814571

A. Limitation of Traditional Cloud Computing: Why Fog Computing Comes?

Cloud computing suffers from substantial yet unsolved challenges such as large end-to-end delay, traffic congestion, processing of massive amount of data, and communication cost. Some of these issues are caused mainly due to large physical distance between cloud service provider's Data Centers (DCs) [7] (such as Amazon Web Services (AWS), Google, ALTUS, Anacondaweb, Apple, Facebook, Matrix, Microsoft, TATA, ChinaTelecom, China Unicom, at&t, and Bell) and End-User (EU). In addition, since these DCs need to operate round the clock, the carbon footprint is an important issue. According to the Natural Resources Defense Council (NRDC), in 2013, U.S. data centers in total used 91 billion kilowatt-hours (kWh) of electrical energy, and the DC electricity consumption is projected to increase to roughly 140 billion kilowatt-hours annually by 2020, the equivalent annual output of 50 power plants [8]. This will cost American businesses $13 billion annually in electricity bills and emit nearly 100 million metric tons of carbon pollution per year. Although some companies like Apple are moving towards more environmental friendly 100 percent renewable DCs [9], [10] with the wind, solar, and geothermal energy, the carbon emission from DCs will dominate on global carbon footprint [11]. In addition, DCs in most of the cloud service providers (e.g., AWS, Google Cloud) are geographically far apart from each other (see Fig. 1), resulting in large DC to EU delay and Quality-of-Service (QoS) degradation, which are not well-suited for the time-critical service requests.

Fog computing: To reduce the burden of DCs in traditional cloud computing, fog computing emerges as an alternative solution to support geographically distributed, latency sensitive, and QoS-aware IoT applications. *Fog computing* was first initiated by Cisco to extend the cloud computing to the edge of a network [12]. The term '*fog*' is used simply because "*fog is a cloud close to ground*", [12], i.e., **F**rom c**O**re to ed**G**e computing [13] enabling refined and better applications or services. Fog computing is a highly virtualized platform [14] that provides computing, storage, and networking services between EU and DC of the traditional cloud computing.

(a) The AWS cloud operates 42 availability zones within 16 geographic regions around the world.

(b) Google cloud platform has 8 regions, 23 zones, over 100 points of presence comprised of hundreads of thousands of miles of fiber optic cable.

(c) Map of metros where at least one edge node is present.

Fig. 1. (a)-(b) Most of the cloud DCs (e.g., AWS and Google cloud) are geographically far apart from each others, resulting large round-trip delay, network congestion, and QoS degradation while satisfying various service requests from EUs. (c) Edge node is more dense compared to cloud DCs with an aim to reduce network burden and to process the computing in a localized manner in order to support delay-sensitive tasks (accessed on 20-March-2017).

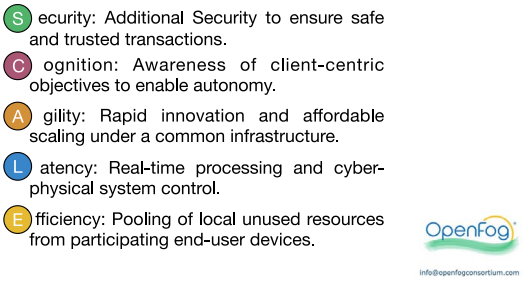

Fig. 2. SCALE is the OpenFog https://www.openfogconsortium.org acronym for Security, Cognition, Agility, Latency and Efficiency.

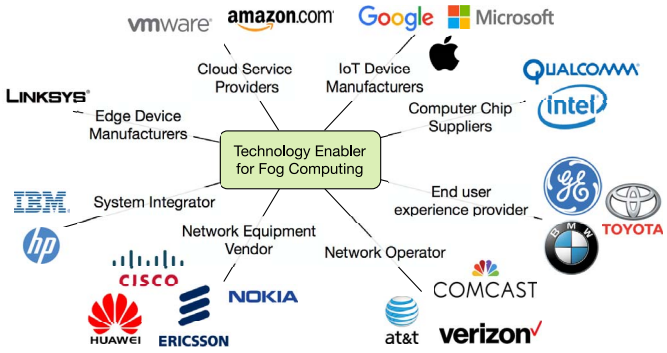

Fig. 3. Technology enabler for fog computing.

Definition: According to [15] 'Fog computing' term is defined as "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. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so."

In another way, fog computing is defined by the OpenFog Consortium [16] as "a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from cloud to Things."

B. Main Features of Fog Computing

Fig. 2 shows some of the fog capabilities such as security, cognition, agility, latency and efficiency, acronym as SCALE, addressed by OpenFog Consortium [17]. Basically, fog computing has the following characteristics [12]:

- low latency and location awareness,
- supports geographic distribution,
- end device mobility,
- capacity of processing high number of nodes,
- wireless access,
- real-time applications, and
- heterogeneity

C. Initiative Towards Fog Computing: Current Market Study

The OpenFog Consortium [17] is a consortium of high-tech giant companies and academic institutions across the world with an aim to standardize and promote fog computing in various fields. This consortium was founded by ARM, Cisco, Dell, Intel, Microsoft Corp., and the Princeton University Edge Laboratory on November 19, 2015. Currently, it has AT&T, FOXCONN, GE, Hitachi, Sakura Internet, ZTE, and ShanghaiTech University as contributing members and 57 members across the North America, Asia, and Europe. By taking the existing industry IoT efforts, OpenFog Consortium workgroups are working towards creating an open architecture for fog computing to enable interoperability and scalability. OpenFog released its Reference Architecture (RA) [18] for fog computing on 13 February 2017. The OpenFog RA is based on eight core technical principles that include security, scalability, openness, autonomy, Reliability, Availability, and Serviceability (RAS), agility, hierarchy, and programmability.

Fog computing is being promoted by several companies such as intelligent edge by Intel [19]. Fig. 3 illustrates a few among many technology enablers for fog computing in various fields discussed by Chiang and Zhang [20]. Some of the examples are EU experience by GE, TOYOTA, BMW, etc., network equipment like switches, gateway by Cisco, Huawei, Ericsson, etc. The current research trends reflect the huge potential of fog computing towards sustainable development in global IoT market.

D. How Fog Computing Works?

A very large number of geo-distributed devices (including EU devices, routers, switches, and access points) form '*mini-cloud*' at the edge of the network. These connected devices manage themselves in a distributed way. The basic idea in fog computing is that instead of always upload/download data to/from core network as in traditional cloud computing, the edge devices which are in proximity, can obtain data from other users through direct link such as Device-to-Device (D2D) communication [21], [22] and adjacent Small Cell (SC) networks [23]. In addition, the edge devices in fog network (FogNet) [15], [24] release some of their resource like computing and storage capacity to support the demands of their neighbors. Only the task that is not well handled by the edge devices are sent to the core cloud part for further processing. As a result, fog computing significantly reduces the computing and routing burden of the cloud.

How fog computing reduces the traffic overhead? If the data of a group of edge entities are highly correlated then there is no need to upload the data from all edge entities, rather the common traffic is uploaded to the cloud from a single edge entity. Similarly, cloud often forwards the highly correlated data to one of the single entity in the group, thereafter, the data are shared among the entities in the group. Thus, the traffic overhead is significantly reduced in fog computing-enabled cloud services. As suggested in [25] and [26] the content of the popular tasks is prefetched in a cache-enabled network with the help of edge computing. Apart from the content fetching, the results of the upcoming computing tasks also are prefetched [27] without prefetching the content of the tasks during the off-peak interval. Therefore, the content and computing caching significantly reduce the burden on access link, thereafter, minimize the overall latency.

E. Basic Difference Between Fog Computing and Traditional Cloud Computing

Although fog computing and cloud computing are interdependent regarding providing computation, storage, and applications, they are different from each other. The main key differences are summarized in Table I. Fog computing extends a substantial amount of data storage, computing, communication, and networking of cloud computing near to the end devices. As fog nodes offload the computation and storage of traditional cloud data centers, fog computing is a viable solution to the resource-constraint devices. Although the end-to-end latency is significantly minimized, the reliability of the services is still better in cloud computing. Another important key difference is that cloud computing tries to optimize resource in a global view, whereas fog computing organizes and manages the *local* FogNet. Also, due to close integration with the front-end intelligence [28]-enabled end devices, fog computing enhances the overall system efficiency, after that improving the performance of critical cyber-physical systems.

F. Related Edge Computing Technologies

Cloudlets [29] also referred as cloudlet-based cyber foraging [30], [31], is one of the first edge computing concepts that brings computing and resource storage closer to the edge. The resource-rich computers with high computation power are placed in a strategic location with EU's vicinity [32]. The computation of the mobile devices is offloaded in one or more virtual machines (VMs). Similar to the WiFi concept, cloudlets provide cloud service to the mobile users instead of proving Internet connectivity as in WiFi [33]. The mobile users can find their preferable cloudlets. Also, the cloudlets exist as a standalone environment since VM provisioning can be supported in cloudlets without the intervention of the cloud. However, as cloudlets are not an integral part of a mobile network, sometimes it becomes hard to fulfill the Quality-of-Service of the mobile devices [34].

In 2014, the Industry Specification Group (ISG) within European Telecommunication Standards Institute (ETSI) integrated edge computing into the mobile network architecture, outlined as Mobile Edge Computing [35]. The purpose of the ISG is to create a "standardized, open environment which will allow the efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor Mobile-edge Computing platforms" [36]. This standardization effort is driven by several leading mobile operators such as DOCOMO, Vodafone, and TELECOM Italia and manufacturers such as IBM, Nokia, Huawei, and Intel [34]. In March 2017, the ETSI has expanded the scope of MEC and after that replaced the term "Mobile" by "Multi-access". The edges of non-mobile networks are also being considered in multi-access edge computing (MEC) [36].

TABLE I
COMPARISON OF TRADITIONAL CLOUD COMPUTING AND FOG COMPUTING

Features	Traditional cloud computing	Fog computing
Computing model	Centralized	Distributed fog nodes are controlled in both distributed and centralized manner
Deployment cost	High due to sophisticated planning	Low, fog enables ad-hoc deployment with or without planning
Resource optimization	Global	Local
Size	Cloud data centers are very large in size	Smaller, however, a large number of small fog nodes form a large fog system.
Mobility management	Easy	Hard
Latency	High	Very low
Operation	Operated by large companies	Often operated by small companies, however large companies can operate depending on the size
Reliability	High	Low
Maintenance	Operated and maintained by technical experts	Generally requires no or little human involvement
Applications	Cyber-domain applications	Support both cyber-domain and cyber-physical applications, most importantly time-critical applications

TABLE II
BASIC DIFFERENCE BETWEEN EDGE COMPUTING TECHNOLOGIES

Features	Cloudlets	MEC	Fog computing
Applications	Mobile offloading	Focus on the applications that are better suited for both mobile and non-mobile network edge	Support a wider range of latency-sensitive applications for resource-constraint end devices
Virtualization	Only depends on VMs	Can use other technologies apart from VMs	Other virtualization technologies can be used
Operational Mode	Can work in standalone mode	Can work in standalone and can connect to the cloud	Cannot work in standalone, need the support of cloud

From the above discussion, it is observed that cloudlets, MEC, and fog computing focus on edge computing, however, there is a significant difference between these technologies. Note that the ETSI is the main driving force for MEC [34], [36], however, fog and cloudlets are driven by research and development, although OpenFog consortium aims at the development of fog architecture and standardization details. Moreover, cloudlets use only VM for virtualization, however, both fog and MEC can use other technologies other than VMs for virtualization [32]. Another main difference is that cloudlet mainly focuses on mobile offloading, and MEC aims to handle the applications that are better suited in either mobile or non-mobile edge networks, however, fog computing overlaps between edge and cloud while satisfying a large number of latency-sensitive applications for resource-constraint end devices [32]. Finally, until today, MEC works only in standalone mode while cloudlets can function in standalone and can connect to the cloud, although there is a lack of detail discussion in literature how to interact between cloud and cloudlets, fog computing needs the support of the cloud for the tasks that might not be performed in the resource-constraint fog layer. Table II summarizes the basic differences between fog, cloudlets, and MEC.

G. Related Surveys and Our Contributions

There are related survey papers [20], [37]–[40] that cover different aspects of fog computing. For example, a brief overview of fog computing as an extension of cloud computing is discussed in [38]. However, this survey is very limited regarding architecture and research challenges in fog computing. Furthermore, Yi *et al.* [39] presented a study that deals with several issues and various metrics like connectivity, reliability, computing, and delay in fog computing. Chiang and Zhang [20] mainly focused on the networking context of IoT and discussed how fog computing could able to solve several research challenges of IoT.

In 2017, Hu *et al.* [41] presented a survey on recent advancement in fog computing, however, resource allocation, service management issues particularly for fog computing are not discussed in detail. Recently, another survey is focused on how cloud and fog computing paradigm handle the research challenges such as knowledge discovery using data analytics with limited computational resources of edge devices to build a sustainable IoT infrastructure for smart cities [40]. Dynamic discovery of Internet objects, multi-protocol support, mobility, context discovery, data analytics and security issues are discussed in the context of fog computing for smart cities. Furthermore, several case studies are presented for smart cities highlighting important open challenges for the realization of fog computing in smart cities. However, algorithmic aspects are not considered in [40].

Most recently, Mahmud *et al.* [37] detailed a taxonomy of fog computing according to challenges and features of fog computing. How related computing paradigms, e.g., mobile edge computing, mobile cloud computing are different from fog computing is mentioned. Also, they studied the fog node configuration with networking devices, various metrics of fog computing. However, the service allocation, resource management and detail performance analysis based on latency, energy consumption, resource sharing, and caching is missing. A more detail analysis is presented in [32] covering on the two different applications domain namely IoT and content delivery networks (CDN) as the basis of driving force to the emergence of fog computing. Several architecture and algorithms are categorized with a prospects of fog computing which is foreseen as a key role in Tactile Internet. However, compared to [32], our survey is different in the organization of resource sharing and research issues in F-RAN architecture. In addition, we have outlined the logical relationship among the different algorithms and approaches in fog computing. In addition, the research challenges and open issues are presented in more critical way.

Review this survey.

Stojmenovic *et al.* [42], [47], discussed the application of fog computing to real scenarios, such as smart grid, vehicular networks, Cyber-Physical Systems (CPS). The security and privacy issues are also briefly presented. More details on security and privacy along with some counter measurements are found in [45] and [48]. Although security and privacy issues in fog computing [42]–[44], [49]–[52] are worthwhile to investigate, in our survey, we do not consider the security and privacy issues since a separate study is essential to review these issues in detail. In recent past, several studies [42]–[46] provide an overview of existing security and privacy concerns, particularly for fog computing.

Table III summarizes the main focus and major contributions of the previous comprehensive surveys on fog computing. While the aforementioned surveys [20], [37], [39], [40], [42]

TABLE III
RELATED SURVEYS ON FOG COMPUTING

Year	Author	Main focus/contributions
2014	Stojmenovic and S. Wen [42]	Security Issues in fog computing.
2015	Yi et al. [39]	Connectivity, reliability, computing, and delay in fog computing.
2015	Saharan and Kumar [38]	Overview of fog computing as an extension of cloud computing.
2016	Chiang and Zhang [20]	Networking context of IoT, how fog can solve the issues for IoT.
2017	Perera et al. [40]	Knowledge discovery using data analytics for smart cities.
2017	Hu et al. [41]	Key technologies for fog computing.
2017	Ni et al. [43]	Security issues in vehicular fog computing.
2017	Ni et al. [44]	Security challenges in fog computing.
2017	Alrawais et al. [45]	Security and privacy issues in fog computing for IoT.
2017	Mukherjee et al. [46]	Security and privacy issues in fog computing.
2017	Mouradian et al. [32]	Comprehensive survey, IoT and CDN as the basis of driving force to the emergence of fog computing.
2018	Mahmud et al. [37]	A taxonomy of fog computing according to challenges and features of fog computing.

have laid a solid foundation for the understanding of taxonomy in fog computing, our survey differs from previous surveys in many aspects.

The following summarizes our key contributions.

1) This article aims at providing a comprehensive survey of fundamental and recent advancement towards fog computing-enabled network architecture.
2) This article summarizes the key research challenges and how state-of-the-art consider these issues in different network applications. The fundamental aspects that should be carefully considered while designing fog computing networks are also presented.
3) Furthermore, this survey highlights some of the open research challenges and discusses the future research directions.

H. Organization

The remainder of this article is organized as follows. Section II states various architecture to support fog computing. Networking, latency, and energy consumption model for fog computing are discussed in Section III. Section IV briefly overviews the fog radio access network (F-RAN) and its research challenges. The various issues in service allocation and resource management in fog computing are discussed in Section V. Section VI presents the applications of fog computing. The challenges and research issues in fog computing are addressed in Section VII. Finally, conclusions are drawn in Section VIII. For the sake of clarity, the organization of this paper is shown in Fig. 4.

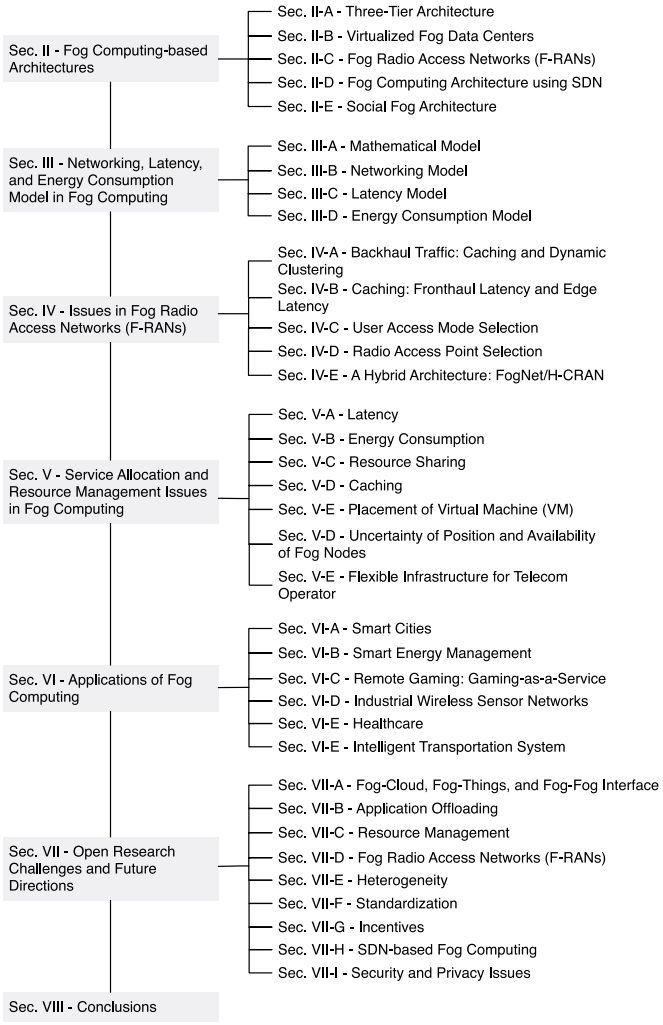

Fig. 4. The organization of this survey.

II. FOG COMPUTING-BASED ARCHITECTURES

To understand how fog computing extends the computing, communication, storage, and service capability at the network edge, we summarize the hierarchical architecture of fog computing as well as software-defined fog architecture, and Fog Radio Access Networks (F-RAN).

The three-tier architecture [53] is one of the basic and widely used architectures in fog computing. Fig. 5 illustrates the architecture. The tiers are discussed as follows:

- *Tier 1–Things/End Devices:* This tier consists of IoT-enabled devices including sensor nodes, EU's smart hand-held devices (e.g., smartphones, tablets, smart cards, smart vehicles, and smartwatch), and others. These end devices are often termed as Terminal Nodes (TNs). It is assumed that these TNs are equipped with Global Positioning System (GPS).
- *Tier 2–Fog:* This tier also termed as fog computing layer. The fog nodes in this layer are comprised of network devices such as a router, gateway, switch, and Access Points (APs). These fog nodes can collaboratively share storage and computing facilities.

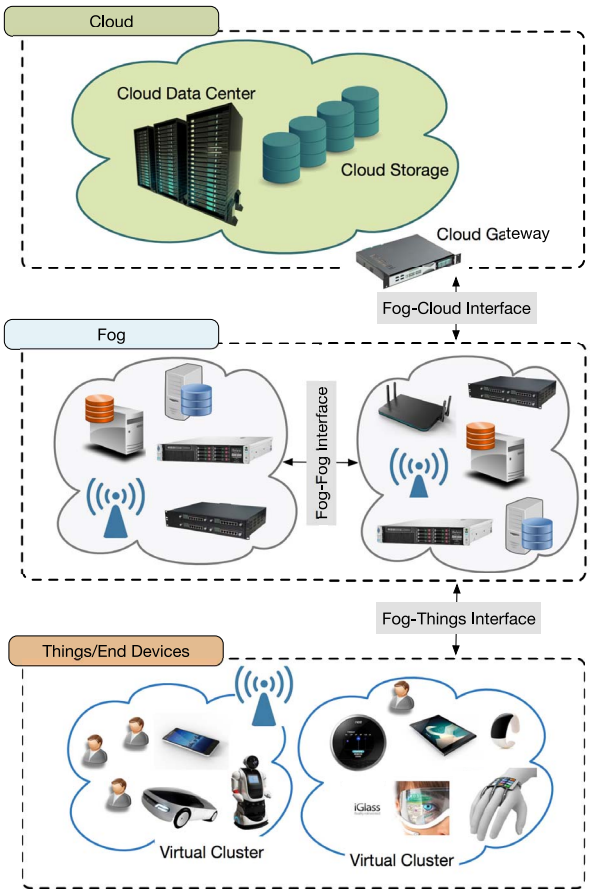

Fig. 5. Three-tier fog computing architecture.

Transport Layer	Sends data to Cloud
Security Layer	Handles security related issues
Temporary Storage Layer	Stores the data temporarily
Preprocessing Layer	Data filtering and trimming
Monitoring Layer	Handles service requests and energy consumption issues
Physical and Virtualization Layer	Contains TNs and Virtual sensor node

Fig. 6. Layered architecture in fog computing.

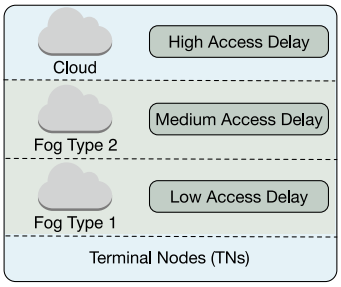

Fig. 7. An architecture of Combined Fog-Cloud (CFC) [54]. Fog second type aims to handle *medium* amount of service request that cannot avail resource in fog first type, however, these requests need not to be executed in the cloud.

- *Tier 2–Cloud:* Traditional cloud servers and cloud DC reside in the top-most tier. This tier has sufficient storage and computing resources.

Layered architecture for fog computing: In addition, Aazam and Huh [14] provide a layered architecture that contains physical and virtualization, monitoring, preprocessing, temporary storage, security, and transport layer as shown in Fig. 6. Physical and virtualization layer mainly contains physical TNs and virtual sensor nodes. The monitoring layer handles the requested task and observes the energy consumption issues of underlying physical devices. Data management related tasks such as data filtering and data trimming are performed in preprocessing layer. The temporary storage layer stores the data for a limited time only. The security-related issues are handled in the security layer. Finally, the transport layer is responsible for sending data to the Cloud.

Different storage and computing capacity in fog nodes: Although the fog nodes significantly reduce the end-to-end delay for the service requests, there may be a situation of different amount of service requests with various delay constraint. For example, some service requests require the low end-to-end delay, however, current resource pool in the fog layer cannot support these requests, nevertheless, these requests do not necessarily to be executed in the Cloud.

To this end, Souza *et al.* [54] suggested a design for Combined Fog-Cloud (CFC) architecture [55], [56]. Fig. 7 depicts the architecture, where the hierarchy of a layer is determined by capacity, vicinity, and reachability to UEs. These tiers are discussed as follows:

- The bottom layer mainly consists of TNs, which can request and offer computing resource to the CFC model.
- The *fog first type* is generally connected to TNs by single hop. This layer is composed of low capacity fog servers with an aim to provide delay-constraint services requests.
- Above this, the *fog second type* with fixed nodes enables collaborative sharing with resource aggregation in the neighborhood area. This layer benefits a medium number of service request with the low delay in a scenario when there are insufficient resources in a lower first layer with a 1-hop connection, however, these service requests are not necessarily to be connected to the cloud server.
- Finally, upper layer contains the traditional cloud servers with enough computing resources, however, at the cost of higher latency to the TNs.

A. Virtualized Fog Data Centers

In a data center, computing and storage devices are interconnected by Data Center Network (DCN) [57], [58]. A DCN is described by the network topology, protocols, routing/switching equipment. A detailed study on electrical element-based, switch-centric, and service-centric topologies for DCN infrastructure can be found in [59]. Clos topology [60], which is one of the widely used switch-centric

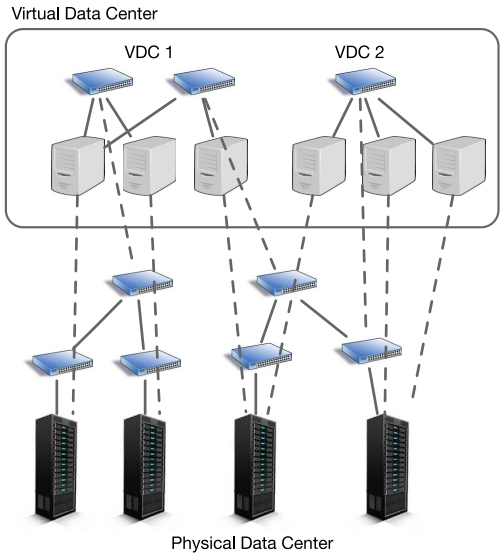

Fig. 8. Data center virtualization.

topologies, has multiple stages of switches. Each switch in a stage is connected to all switches in the below stage. Fat-tree topology [61], [62] that is one of the special types of Clos topology has a tree-like structure. This topology is well-suited for the intra data center networks. More details can be found in [61] and [62]. Note that in a fat-tree topology, each edge switches are directly connected to an EU.

Data center visualization creates multiple virtual networks (VNs). These VNs can be implemented independently. Since VNs are logically separated from each other, performance isolation can be performed. Also, deployment of the new application becomes easier in virtualized DCNs. Basically, in a virtualized data center, hardware resources including servers, routers, switches, and links are virtualized. The hypervisor, a firmware, divides the physical equipment into independent and isolated virtual instances. Fig. 8 shows how several virtual DCs (VDCs) are deployed over a virtualized data center. A VDC consists of virtual machines (VMs), virtual switches, and virtual links. Therefore, VDC is a logical instance of the virtualized data center. Note that virtualization is introduced in network virtualization level, which is one of the layers in the network stack.

Some of the requirements of traditional data centers such as performance isolation, high server utilization, and low operational cost can be fulfilled by the server virtualized technologies, such as VMware [63], Xen [64]. However, other limitations, e.g., performance isolation, security issues, application deployment, management flexibility cannot be easily addressed by above technologies. Data center virtualization [57], [65] aims to improve flexibility and efficiency of the computing to satisfy Data Service Subscribers' (DSSs') requirement. Massive data center operators create a virtualized server based on different requirements of DSSs'. As only one server in the data center virtualization can support multiple DSSs, capacity utilization and energy efficiency can be dramatically improved. However, since massive data centers are

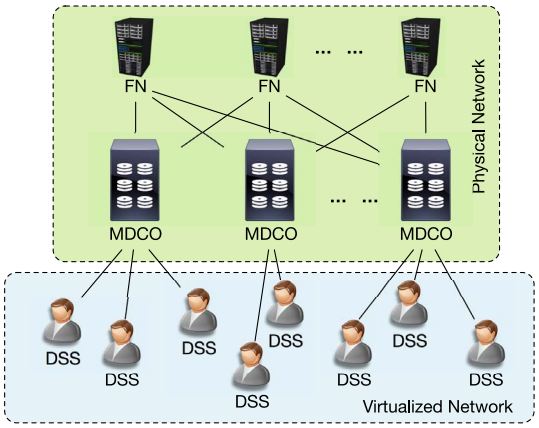

Fig. 9. Data center networks with fog computing.

What do they mean by cost, herein? Comparison between traditional Data centers and the Fog computing architecture.

far away from DSS, these massive data centers suffer from transmission delay and cost while satisfying DSSs' requirements. To solve these issues, fog computing enables to add multiple virtualized edge data centers to offload the service from these traditional massive data centers. Several architectures like in [66], EdgeCloud [67], micro-data center [68], and nano Data Centers (nDCs) [69] were proposed to provide better service to DSS while offloading the massive data centers. A 3-tier architecture was proposed in [70] where FNs reside in upper layer, Massive Data Center Operators (MDCOs) are in middle layer, and DSSs are in bottom layer as shown in Fig. 9. FNs share the computing and storage resources with MDCOs, after that MDCOs provide service to DSSs with low latency. It was also assumed that each of the MDCOs could support different service with SecondNet [71] topology. A more detail can be found in [72] and [73].

B. Fog Radio Access Networks (F-RANs)

The Spectral Efficiency (SE) can be significantly improved in Cloud Radio Access Networks (C-RANs) [74]–[76], where interference management is more efficient due to the virtualization of baseband processing of Remote Radio Heads (RRHs) (or Radio Units (RUs) [77]) at a Centralized Processor (CP). Nevertheless, F-RAN [78] enhances a C-RAN by enabling the RRHs, called *enhanced* eRRHs (eRRHs), with caching and signal processing capabilities. By introducing *local* cache to the eRRHs in F-RANs [78] can significantly reduce the traffic overhead and latency because these popular files can be fetched from eRRHs rather than CPs via a fronthaul.

In C-RAN architecture, eRRHs can be separated from eNodeBs and eRRHs are massively deployed to extend the coverage of eNodeB. Compared to C-RAN, a conceptual architecture with a deployment of High Power Node (HPN), called as heterogeneous-CRAN (H-CRAN), is addressed by Peng *et al.* [81] and Lei *et al.* [82]. Since these HPNs can exchange information for joint resource scheduling, the inter-cell interference can be mitigated. A new RAN architecture is suggested in [79] and [80] to combine the advantages of both H-CRAN and FogNet. To fully utilize the network and

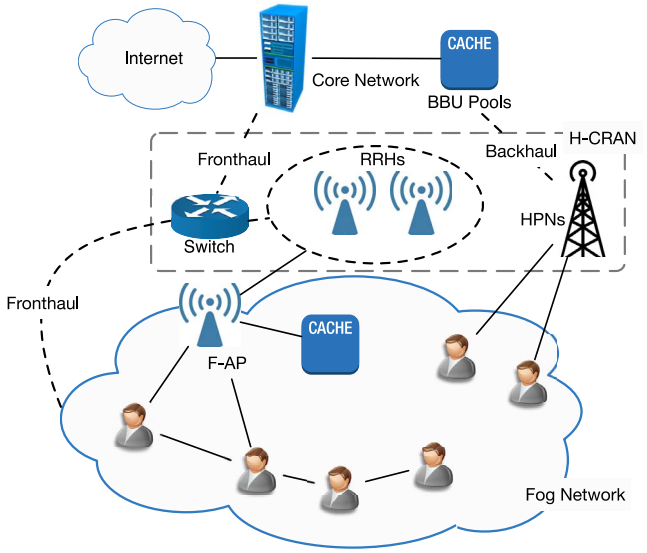

Fig. 10. Architecture of FogNet/H-CRAN [79], [80]. Cloud-part, which is composed of eRRHs and HPN, provides the ubiquitous connection service to all EUs. The devices can exchange data locally or connect to the cloud via F-APs.

radio resource from the viewpoint of cache, mobility, and access control, this proposed mechanism switches between H-CRAN and FogNet. Basically, H-CRAN globally optimizes the resource allocation, whereas FogNet focuses the information exchange and computation at the edge of the network [80]. Fig. 10 illustrates the architecture of H-CRAN/FogNet. In this hybrid model, the data can be retrieved in the following three-way: 1) directly from other users, 2) cached at Base-Band signal processing Unit (BBU) pool via nearby eRRHs, and 3) from cloud network. Integration with short-range communications, e.g., mmWave and licensed assisted access would be one of the possible future directions.

C. Fog Computing Architecture Using Software-Defined Networking (SDN)

Software-Defined Networking (SDN) [83], [84] is an emerging solution that provides a flexible way to update and reconfigure the network. The main idea of SDN is to separate the control plane and the data plane physically. The sensor node does not have any task-decision capabilities, rather is being controlled by the instructions from a centralized controller. Therefore, these sensor nodes are easy to control. Also, re-tasking or reprogramming is very simple without any significant change in hardware of the deployed sensor nodes. OpenFlow [84], an open protocol, which is a standard interface between control and data-plane, allows reprogramming and update in the FlowTables.

Generally, the SDN controller handles the packet forwarding and other networking function [83]. These controllers interact with switches mainly via TCP connection. However, in SDN, the delay between controller and switch becomes one of the limiting factors. Distributed controller placement seems to be a solution while managing the data flow in a localized manner.

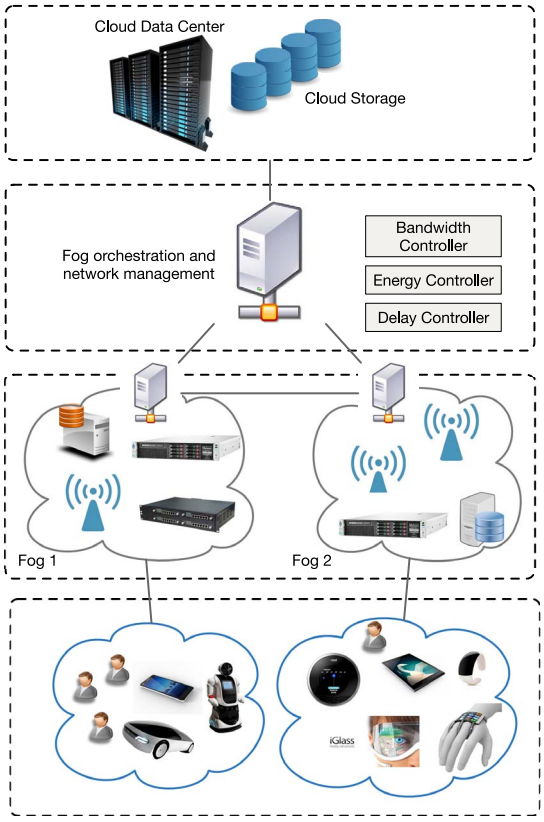

Fig. 11. SDN-based fog computing architecture.

However, the number of controller increases due to one controller per network, which further increases the one-hop delay. To reduce the burden on control-plane, several approaches are suggested. A switch called '*authority switch*' [85] is proposed to offload some functionality of s controller by using pre-installed rules. To reduce infrastructure overload, edge nodes are designed towards data offloading [86] in Delay-Tolerant Networks (DTNs).

Although fog computing is a viable solution for the latency-sensitive task, the available resource can vary as the dynamic nature of infrastructure, mobility issue of TNs, and traffic distribution, that results in an outage of available fog computing resources. Therefore, it may be beneficial if some of the latency-aware fog tasks can be transferred to the cloud if there is not enough resource in the fog layer. On the other hand, SDN benefits from complete knowledge of network state which allows distribution of latency-aware fog tasks. Also, dynamic QoS policy deployment is required to handle fog services with different QoS. Therefore, software-defined QoS provisioning has to be realized in fog computing. Basically, with fog computing, the computation and transmission burden on SDN controller can be reduced. At the same time, the scalability and resource management can be improved in fog networks implemented by SDN [87], [88].

The basic architecture of fog computing with SDN is shown in Fig. 11. The main difference between the traditional 3-tier fog architecture and SDN-based fog architecture is the fog-SDN controller to support dynamic QoS [89]. The SDN controller layer is responsible for defining fine-grained QoS

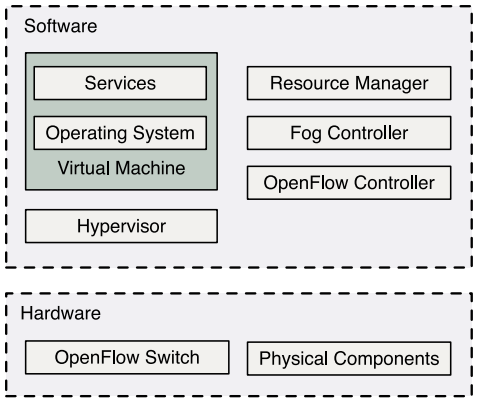

Fig. 12. The components of fog-SDN controller.

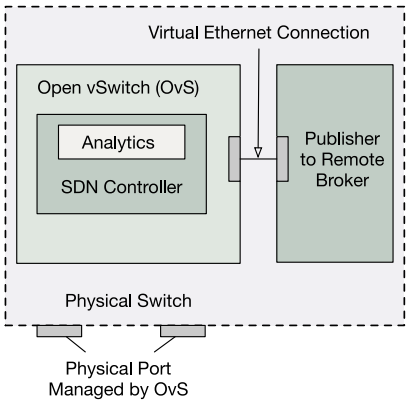

Fig. 13. SDN-based fog (switch) node [93]. Functionality of the broker node is enabled in edge-switch, where the SDN switch-side is custom modified to integrate SDN-controller within switch hardware.

provisioning based on the attributes and the state of data gathered from fog node [89], [90]. To enable fog computing framework for the SDN-based system, the SDN controller also needs to support virtualization for allowing fog service. Therefore, a hypervisor, a low-level middleware, is required to be implemented at this SDN controller to support the abstraction of VM on them [91]. Fig. 12 illustrates the necessary hardware and software components of the fog-SDN controller. Mainly, these SDN controllers play as fog orchestration and resource manager.

Baktir *et al.* [92] presented a comprehensive survey on the benefits of SDN in edge computing. The fog computing architecture is implemented at edge switches using SDN [93]. Although, due to the centralized architecture design of SDN, it is not easy to apply fog computing at switches. As a first step, the controller functionality is integrated at edge switches to perform discrete and distributed computation. Message Queuing Telemetry Transport (MQTT) [94], which is a content-based publisher-subscriber model, is considered as a candidate IoT protocol for implementation. From testbed, it is shown that fog node delivers high throughput compared to the traditional client and end-to-end setup.

1. Distribution of fog tasks: Borylo *et al.* [95] suggested that in an optical network, fog traffic can be handled by decreasing average light path length with a slight cost of blocking probability and carbon footprint. In this paper, fog traffic intensity is modeled as Markov Modulated Poisson Process (MMPP) to highlight the uncertainty and bursty nature of this traffic. The proposed latency-aware policy is designed to be adjusted based on the current network condition and expected user traffic. As the proposed policy takes small input data, this solution can be easily implemented in SDN controller. This work is a very early step towards SDN-related combined fog and cloud computing networks. However, optimal DC placement is one of the future research directions. Furthermore, harnessing the SDN, a fog-cloud interoperation is presented in [88] where an SDN controller acts as a bridge between fog and cloud with an aim to improve quality of experience and optimize the network resources. A network-level virtualization, called OpenPipe, integrated with SDN is proposed in [87].

2. Storage, computing, and networking resources of switches: An SDN-based auto-docker framework is suggested in [96] to automatically identify and dock/undock application without EUs intervention. This proposed framework aims to efficiently handle the storage, computing, and networking resources of switches. This approach provides more flexibility to SDN framework by implementing them close to EUs. As shown in Fig. 13, there are main two essential components of Open vSwitch and docker management application. The Open vSwitch crates virtual bridge between individual applications and contains an in-house controller that enables packet forwarding decision. Docker manager application is connected to Open vSwitch via virtual Ethernet connection. The proposed framework manages several features like UE's application request processing, flow initiation, and a pool of applications. Finally, the docking/undocking functionality was successfully demonstrated with two different application MQTT and DTN. To support docker capability using Internet cloud would be an interesting extension of this framework.

Table IV summarized the background and focus of the above architectures.

III. Networking, Latency, and Energy Consumption Model in Fog Computing

The mathematical model of fog computing is required to understand various components and their behavior in the fog layer. As a first step, with the assumption as in [97], Sarkar and Misra [98] introduced a simple yet useful mathematical model of fog computing system as follows.

A. Mathematical Model

The main components in fog computing are discussed as follows.

1) End-User Node: These nodes are often called as TNs (or IoT-enabled devices). Each of these EU nodes has at least six tuple as: $\mathcal{T} =< EU_{\mathtt{ID}}, \mathtt{Status}, \tau_i, \mathcal{L}, \mathcal{H}, \mathcal{I}(q) >$, where $EU_{\mathtt{ID}}$ refers to the ID of the EU nodes, $\mathtt{Status}$ denotes the current status of the EU nodes and takes the values from $\{0, 1\}$, where $\mathtt{Status} = 0$ refers to idle state and $\mathtt{Status} = 1$ denotes active state of the EU node, τ_i indicates the event type and takes the value from the event set $\tau = \{\tau_1, \tau_2, \tau_3, \ldots, \tau_p\}$ with total p distinct events, $\mathcal{L}$ is the geographic location with

TABLE IV
SUMMARY OF FOG COMPUTING-BASED ARCHITECTURE

Architecture	Background	Focus/Contribution
Three-tier architecture [14], [42], [47], [53]	• Basic architecture of fog computing.	• To extend the computing and storage facilities towards network edge and to reduce burden on the cloud DCs.
Four-tier Combined Fog-Cloud (CFC) architecture [54]	• Some service requests require very low end-to-end delay, however, current resource pool in the fog layer cannot support these requests, nevertheless, these requests do not necessarily to be executed in the cloud.	• The hierarchy of a layer is determined by capacity, vicinity, and reachability to EUs. • Two fog layers are suggested. • The fog second layer supports the service requests when there is not enough resource in fog first layer.
SDN-based fog computing [93], [93], [96]	• Separation of data and control plane. • Integrates controller functionality in edge-element, i.e., edge-switch. • Communicates with both fog and cloud management software.	• Improved controller-switch delay performance [93]. • Effectively manages storage, computing, and networking resources of the edge-switch [93], [96]. • Handles the Fog-related traffic with an aim to reduce latency and carbon footprint [95].

time stamp, $\mathcal{H}$ field denotes the hardware and software details of the node, frequency of operation, sampling rate, and hardware used for wireless transmission, and $\mathcal{I}(q)$ presents the ID of the application instances.

2) Virtual Cluster: The EU nodes form a virtual cluster, also termed as *mini-cloud* [37], that corresponds to the logical boundary of these EU nodes. Mathematically, these virtual cluster is defined as $\mathcal{V} =< V_{\mathrm{ID}}, \mathcal{T}(u), R, F_{\mathrm{ID}} >$, where V_{ID} refers to the ID of the virtual cluster, R denotes the region under the virtual cloud, F_{ID} refers to the physical fog instance to which this virtual cloud is mapped, and $\mathcal{T}(u)$ is the set of EU nodes under the virtual cluster. Note that $|\mathcal{T}(u)|$ changes as mobile EUs dynamically join or leave from any virtual cluster.

3) Fog Computing Instances: It can simply be expressed as $\mathcal{F} =< F_{\mathrm{ID}}, C_{AP}, \mathcal{D}(v) >$, where F_{ID} is the ID of fog computing instances, C_{AP} is the access point through which the fog instance is connected to the cloud, and $\mathcal{D}(v)$ stores the device IDs for all fog computing devices.

4) Fog Computing Devices: Fog computing devices have at least these tuple as follows: $\mathcal{D} =< D_{\mathrm{ID}}, \mathtt{Type}, \mathtt{Spec} >$, where D_{ID}, `Type`, and `Spec` denote device ID, device type (such as router, gateway, processing or storage unit), and hardware- and software-related information, respectively.

Basically, the mapping function from the set virtual cluster $\mathcal{V}$ to fog computing instances $\mathcal{F}$, i.e., $\tilde{\mathcal{V}} \rightarrow \tilde{\mathcal{F}}$ is injective, whereas the mapping function from the EU node set $\mathcal{T}$ to fog computing instances $\mathcal{F}$, i.e., $\tilde{\mathcal{T}} \rightarrow \tilde{\mathcal{F}}$ is many-to-one.

B. Networking Model

Let N be the total number of EUs. As these EUs are often grouped into a virtual cluster, assume that $EU_i, \forall i = 1(1)V$ is the number of EUs that are grouped in the ith VC, where $|\mathcal{V}| = V$ is the total number of virtual clusters. The data generated by these terminal nodes are transmitted through the edge gateway to the cloud. The set of edge gateway is denoted as $|\epsilon| = E$. Both the data and service requests are forwarded through the link (e, f) between virtual cluster and the fog computing instances, where $\forall e = \epsilon$ and $\forall f \in \mathcal{F}$. The data generated by the virtual clusters are transferred to the fog instances though bandwidth-limited link (e, f). The data generated by the EUs are either locally processed in the fog computing instances or redirected to the cloud for storage and processing purposes. The fog tier routing variable that represents the uplink path $v_i \rightarrow e \rightarrow f$ for the data from the v_i virtual cluster to the fog instances is defined as $X^{\mathrm{fog}}_{v_i,e,f}$. If there are insufficient resources to process the data or if the data are required to be stored in cloud layer for further analysis, then the cloud tier routing variables is defined as $X^{\mathrm{cloud}}_{f,g,c}$ with the path $f \rightarrow g \rightarrow c$, where the link between fog instances and the fog gateway, i.e., $(f, g), \forall f \in \mathcal{F}, \forall g \in \mathcal{G}$, where $\mathcal{G}$ is the set of fog gateway, is generally bandwidth constrained. At the same time, the link between cloud DC and the fog gateway also has bandwidth constraint. However, the data migration between cloud DC avails very high bandwidth transfer [53]. The $X^{\mathrm{fog}}_{v_i,e,f}$ and $X^{\mathrm{cloud}}_{f,g,c}$ variables indicate the amount of generated data that generate from virtual clusters reach to the fog and thereafter transferred to the cloud, respectively. Also, these variables are used to estimate the service latency and the power consumption [53] in fog computing.

C. Latency Model

Several authors [55], [56], [99] studied the latency issues in fog computing. However, a few of them considered a generalized approach to evaluate the overall latency in fog computing. Although, the computing time that depends on many factors, to simplify the analysis, is represented as a *linear* function [99] for a workload n

$$T_{\mathrm{computing}}(n) = \rho(n) + \varrho, \tag{1}$$

where ρ and ϱ are the gradient and constant of the computing function. Assume that each F_j fog node executes n_j workload at a certain time with maximum workload capacity $n_j^{\max}$. The transmission delay D_{S_i,F_j} is the round-trip-time between S_i and F_j excluding the computing time in a fog node. It is also assumed that a source node can avail any one of fog nodes in the network. Thus, the total latency including computing and round-trip time D_{S_i,F_j} due to workload n_j at a fog node F_j is expressed as $T_{\mathrm{total}}(w_k) = D_{S_i,F_j} + T_{\mathrm{computing}}(n_j + w_k)$, if the workload w_k is accepted by F_j and 0 otherwise.

However, the above assumption in [99] is too simple without considering other factors. A useful mathematical approach to calculate the latency is presented in [100]. Assume that the

underlying physical devices send x_i packets per second to the ith fog node. In general, all of the tasks cannot be offloaded to either cloud or neighboring fog node due to many issues like user privacy and real-time operation. Thus, a fraction of the task x_i, i.e., $\lambda_i = \alpha_i x_i$ is locally computed by the ith fog node, where α_i is the fraction of a unit task computed by the ith fog node. Due to the limited resource compared to the cloud DC, the ith fog node offloads the computing tasks to other fog nodes or cloud. A number of tasks that are offloaded from the ith fog node to the fog node $j \in \mathcal{J}$, where $\mathcal{J}$ is the set of available neighbor fog nodes, will be $\lambda_{ij} = \alpha_{ij} x_i$, where α_{ij} is the fraction of unit task that is offloaded to the jth fog node. Thus, the fog layer computes the task $(\alpha_i + \sum_{j\in\mathcal{J}} \alpha_{ij}) \times x_i$. The rest of the tasks is offloaded to the cloud. Therefore, the total computation delay consists of the following factors [100]:

• *Transmission queue:* The tasks that arrive from the EUs to the ith fog node has to wait before they are distributed to the other fog nodes or cloud. Assuming the task arrival rate as Poisson distribution [56], the latency due to the transmission queue when a task of a size K bit is transmitted from the ith fog node the jth fog node is modeled as

$$T_j(\alpha_{ij}) = \underbrace{\frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij}-\lambda_{ij})}}_{\text{waiting time at fog node } i} + \underbrace{\frac{1}{\mu_{ij}}}_{\text{transmission delay}}, \tag{2}$$

where μ_{ij} is the service rate for the task that are transmitted from the fog node i to the fog node j (or cloud using a base station) and is expressed as

$$\mu_{ij} = \frac{1}{K} BW \log\left(1 + \frac{g_{ij} P_i}{BW \times N_0}\right), \tag{3}$$

where BW is the channel bandwidth, P_i is the transmission power of the ith fog node, N_0 denotes the noise spectral density, and g_{ij} is the channel gain between the ith and jth fog nodes. In addition, the delay due to the transmission queue for the offloaded task to the cloud using a base station c becomes

$$T_c(\alpha_{ic}) = \frac{\lambda_{ic}}{2\mu_{ic}(\mu_{ic}-\lambda_{ic})} + \frac{1}{\mu_{ic}} \tag{4}$$

• *Computation queue:* When a task arrives at the destination fog node j, the task has to wait due to the previous task completion. In addition, the jth fog node not only receives the tasks from the ith node, however also from other fog nodes. With an assumption that the task arrival rate at the jth fog node follows Poisson process [101], the computation latency in the fog node j is expressed as

$$S_j(\alpha_{ij}) = \underbrace{\frac{\lambda_{ij}}{2h_j(h_j-\lambda_{ij})}}_{\text{waiting time at fog node } j} + \underbrace{\frac{1}{h_j}}_{\text{fetching delay}} + \underbrace{d_j}_{\text{task computation}}, \tag{5}$$

where h_j depends on the overall hardware of the jth fog node and $d_j = \lambda_{ij}/c_j$ is the computation time to execute the task, where the constant term c_j is proportional to CPU clock frequency of the fog node j. Similarly, when a task is locally processed in the ith fog node, the computation latency becomes

$$S_j(\alpha_i) = \frac{\lambda_i}{2h_i(h_i-\lambda_i)} + \frac{1}{h_i} + d_i, \tag{6}$$

As the computation and storage capability of the cloud are significantly higher than a fog node, the waiting and fetching time can be ignored concerning the computation time. Thus, the computation delay for the tasks in cloud is $S_c(\alpha_c) = d_c = c_c \lambda_c$.

Finally, the total latency when a task is transferred to the cloud becomes

$$D_c(\alpha_c) = T_c(\alpha_c) + S_c(\alpha_c). \tag{7}$$

Similarly, when the task is offloaded to the jth fog node, then the total latency is

$$D_j(\alpha_{ij}) = T_j(\alpha_{ij}) + S_j(\alpha_{ij}). \tag{8}$$

As there is no transmission queue for a task which is locally computed at the ith node, the total delay comprises of only $S_i(\alpha_i)$.

In general, considering the latency issue, the objective function for task distribution as well as resource sharing becomes [100]

$$\arg\min_{\mathcal{J},\boldsymbol{\alpha}} \quad \max\left(D_i(\alpha_i), D_c(\alpha_c), D_j(\alpha_{ij})\right) + \eta(J+1) \tag{9a}$$

$$\text{subject to:} \quad \alpha_i + \alpha_c + \sum_{j\in\mathcal{J}} \alpha_{ij} = 1; \tag{9b}$$

$$(\alpha_i, \alpha_c, \alpha_{ij}) \in [0,1] \ \forall j \in \mathcal{J}; \tag{9c}$$

where $\boldsymbol{\alpha} = [\alpha_i, \alpha_c, \alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{iJ}]$ is the task distribution vector and $\eta(J+1)$ is the delay due to transmission queues for total J neighboring fog nodes and the cloud.

Latency in cache-enabled fog networks: In cache-enabled fog network, the popular contents are prefetched to reduce the burden on access links. In addition, followed by proactive computing [26], [102], the results of popular tasks can also be cached in advanced. To find the computation latency in cache-enable fog network, a mathematical model is introduced in [27]. Basically, this model considers a fog network that consists of a set of edge computing nodes, i.e., cloudlets and a set of uniformly and randomly distributed EUs. Each of the cloudlets has computing and storage capability of c_e and s_e, respectively. The EUs intend to execute a set of tasks, and each task requires κ CPU cycles per bit of data and task data size follows an exponential distribution with a mean L_a. In fog computing, all the tasks are not always be offloaded due to several reasons. The total computation delay of a task a that is either offloaded to cloudlet or locally processed is expressed as

$$D_a(t) = \mathbb{1} \times D_{ea}^{\text{fog}}(t) + \left(1 - \sum_{e\in\mathcal{E}} \mathbb{1}\right) D_{ua}^{\text{local}}(t), \tag{10}$$

where $\mathbb{1} = 1$ if task a is distributed to cloudlet e or 0, otherwise, $D_{ea}^{\text{fog}}(t)$ and $D_{ua}^{\text{local}}(t)$ are the total computing delay for the tasks that are offloaded to the cloudlet and locally processed, respectively.

The local computing time for the task a that is computed locally by EU u is expressed as

$$D_{ua}^{\text{local}}(t) = \underbrace{\frac{\kappa L_a}{c_{\text{local}}}}_{\text{computing delay}} + \underbrace{W_{ua}^{\text{local}}(t)}_{\text{local queue delay}} + \underbrace{\tau_u}_{\text{local processing delay}}, \tag{11}$$

where c_{local} is the local computing capability.

When a task is offloaded to a cloudlet, the delay occurs mainly due to data transmission time, cloudlet computing time, queuing, and processing time. The delay is expressed as

$$D_{ea}^{\text{fog}}(t) = \left(\underbrace{\frac{\kappa L_a}{c_e}}_{\text{computing delay}} + \underbrace{\frac{L_a}{R_{ue}}}_{\text{transmission delay}} + \underbrace{W_{ea}^{\text{fog}}(t)}_{\text{local queue delay}} \right) \times (1 - \mathbb{C}) + \underbrace{\tau_e}_{\text{cloudlet processing delay}}, \tag{12}$$

where R_{ue} is the uplink data rate and $\mathbb{C} = 1$ if the computing result of the task a is already cached in the eth cloudlet or 0, otherwise. Note that the downlink latency can be ignored with respect to the uploading and computation time in the cloudlet.

D. Energy Consumption Model

Energy consumption is one of the significant performance metrics for fog computing. As fog computing emerges, tiny servers known as "nano Data Centers (nDCs)" [69] become very popular for hosting and distributing contents in a peer-to-peer (P2P) way. Although some studies [103] suggested that P2P content distribution consumes more energy compared to centralized DCs, nDCs are the viable solution to save a significant amount of energy by pushing content closer to the EUs. Jalali *et al.* [104] derived a comprehensive energy consumption model for content distribution in nDCs. Although existing literature differs in models of energy consumption, this energy consumption model can be useful to extend in fog computing scenarios.

Energy consumption model in nDCs: The network types of equipment are broadly categorized as follows: a) shared by many users and b) shared by a few users or a single user. For heavily shared equipment, a *flow-based* energy model was suggested, whereas, for the equipment, which is not shared by many users, a *time-based* energy consumption model was used for network equipment.

- *Flow-based energy consumption model:* The equipment that is shared by the many users includes routers, switches, etc. Since the energy consumption in a cloud service depends on the proportional allocation of the equipment's power over all the flows through the equipment, power consumption in a typical network equipment with C capacity (in bit/second) is modeled as a linear form as [105] and [106]

$$P(C) = P_{\text{idle}} + \frac{C(P_{\max} - P_{\text{idle}})}{C_{\max}}, \tag{13}$$

where P_{idle}, $P_{\max}$, and $C_{\max}$ are the idle power, maximum power, and maximum capacity supported by the network equipment, respectively. Since P_{idle} consumes a significant amount of energy, this term cannot be ignored while calculating the energy consumption of a service.

The additional energy consumption of kth service that transfers B bit across the network is expressed as:

$$E_{\text{k-flow}} \approx E_{\text{b-flow}} \times B, \tag{14}$$

where $E_{\text{b-flow}}$ denotes the incremental energy per bit due to increase in traffic and is written as

$$E_{\text{b-flow}} = \frac{\Delta P}{\Delta C} \approx m\left(\frac{< P_{\text{idle}} >}{U < C_{\max} >} + < E_b > \right), \tag{15}$$

where the service under consideration adds incremental capacity ΔC, m represents the average number of network nodes in the service path, $< C_{\max} >$ is the average maximum capacity of the network elements, E_b is the energy per bit, and U is the utilization of the elements set by the network operator.

- *Time-based energy consumption model:* Time-based energy consumption model depends on the amount of time that the user equipment or nDC spends to provide access to the services. Consider that the device is actively using a service, say k, during the $t_{a,i}$, $\{i = 1, 2, \ldots, n\}$ time-slot and being idle at $t_{s,i}$, $\{i = 1, 2, \ldots, n\}$ time-slot. Thus, the total time nDC spends becomes $T_{\text{Total}} = \sum_{i=1}^{n}(t_{a,i} + t_{s,i})$, and total active time T_{active} is $\sum_{i=1}^{n} t_{a,i}$.

Therefore, the energy consumption for TNs becomes

$$E_{\text{TN}} = P_{\text{idle}} T_{\text{Total}} + \int_{T_{\text{active}}} (P(t) - P_{\text{idle}}) \mathrm{d}t, \tag{16}$$

where $P(t)$ and P_{idle} are the total and idle-state power consumption of the device. If the device serves more than one service, then the energy consumption of one specific service is calculated as follows: 1) incremental energy consumption due to running the specific service, $E_{\text{inc},k}$ and 2) idle power allocated to running the kth service, $E_{\text{idle},k}$. The total energy consumption of the kth service is $E_k = E_{\text{inc},k} + E_{\text{idle},k}$. Let define a co-efficient $\phi = t_{\text{idle}}/t_{\text{active}}$, i.e., the ratio of the idle time to the active time of the device. When $\phi = 0$, then the nDC is fully utilized. Using ϕ, (16) can be rewritten as

$$E_{\text{TN}} = P_{\text{idle}}(\phi + 1) T_{\text{active},k} + \int_{T_{\text{active},k}} (P(t) - P_{\text{idle}}) \mathrm{d}t, \tag{17}$$

where $T_{\text{active},k}$ is the active time of the kth service.

Since it is not very convenient to monitor DC's time allocation to each user, the ϕ-factor is not applicable in centralized DCs. From above discussion, it is observed that the ϕ-factor mainly represents the amount of unshared resource in nDCs. As nDCs are too small to highly share the resource among many workloads, time-based energy consumption model with the ϕ-factor, whereas centralized DCs that are highly shared over more than thousand workloads, a shared-equipment flow-based energy consumption model is applied.

IV. Issues in Fog Radio Access Networks (F-RANs)

Spectral Efficiency (SE) and Energy Efficiency (EE) become important design metrics for the evolution towards 5th Generation (5G). To increase SE, dense and SC architecture were successfully introduced. The heterogeneous network, abbreviated as HetNet [23], mainly consists of femtocell

TABLE V
COMPARISON OF VARIOUS RADIO ACCESS TECHNOLOGY

RAT	Feature	Advantages
UMTS [108]	Each RNC controls NodeBs.	Resource management is performed in RNCs
LTE/LTE-A	Radio resource optimization also considers NodeBs	Joint resource scheduling/optimization among multiple eNodeBs
C-RAN [74]–[76]	BBU and eRRHs are separated from eNodeB	Coverage of eNodeB is extended. In addition, BBU and eRRH can be massively deployed.
H-CRAN [81], [82]	High power nodes are massively deployed	Co-ordination of eRRH and HPNs is improved. Along with BBU/RRH, relay nodes and HPNs also participate in resource optimization.

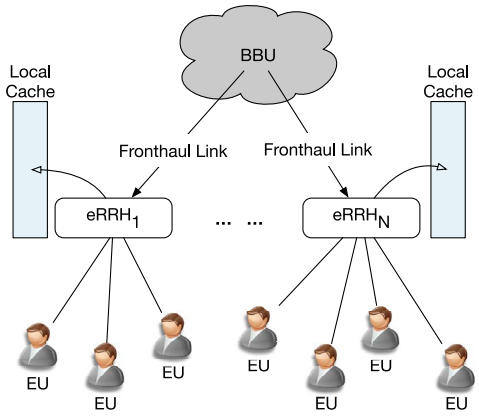

Fig. 14. In F-RAN, BBU in the cloud performs the baseband processing whereas eRRHs have both caching and signal processing capabilities. Pre-fetching and delivery are common phases in a typical cache-enabled systems.

and pico-cell underlaying or overlaying macro-cells. Recently, Millimeter Wave (mmWave) communication [107] at 28 GHz and 60 GHz has shown a promising approach to overcome spectrum shortage. The C-RAN architecture [74]–[76] is originated from Universal Mobile Telecommunications System (UMTS) [108], where each Radio Access Controller (RAC) controls NodeBs which are the responsible for physical signal transmission. Table V summarizes the main features and advantages of the recent Radio Access Technologies (RATs). To further increase the SE in C-RAN, RRHs are densely deployed, however, the backhaul, i.e., the path connecting the base stations to the CP, becomes the main limiting factor for overall network performance. In addition, the capacity of fronthaul links between BBU and RRHs limits the performance of C-RANs. As shown in Fig. 14, eRRHs are equipped with caching and signal processing capabilities. The popular files can be fetched from eRRHs rather than CPs via the backhaul. Thus, the *local* cache in F-RANs [78] can significantly reduce the traffic overhead and latency. In this context, primary research challenges and proposed approaches in F-RAN are shown in Fig. 15. In the following, some of the challenges in F-RAN are discussed in detail.

A. Backhaul Traffic: Caching and Dynamic Clustering

File caching becomes one of the critical issues in the F-RAN. Assuming all requested files are available at CP, these files are indexed in CP based on the popularity index. The popularity of a file is modeled as Zipf distribution [77], [109], where the probability of the file being requested is expressed as

$$P(f) = \frac{f^{-\alpha}}{\sum_{j=1}^{F} j^{-\alpha}}, \quad f = \{1, 2, \ldots, F\}, \tag{18}$$

where α is the skewness of the distribution. The file with smaller index has a larger probability of being requested. The fundamental limits of several caching schemes are discussed in [110]. These caching schemes are broadly divided into *uncoded* and *coded* caching. In uncoded caching, the complete files are being cached, whereas different fractions of files are stored in the different cache using Maximum Distance Separable (MDS) code. Each eRRH caches the same popular file until its memory is full. Afterward, these eRRHs can collaboratively serve the EUs' requests without consuming backhaul resources. Nevertheless, the uncached file creates a burden to the backhaul since uncached files need to be fetched from the CP. In this scenario, the number of eRRHs plays an important role. For example, if more eRRHs are involved for uncached file sharing, then power consumption will be less. However, the burden on backhaul will be increased. On the other hand, the backhaul burden will decrease when less number of RUs are involved. However, due to less cooperation, the power consumption will increase. Fig. 16 illustrates the above trade-off. This kind of trade-off between total power consumption and total backhaul capacity is discussed with both uncoded [111] and coded [112] caching process in F-RAN.

Furthermore, to minimize power consumption by considering individual backhaul capacity, an algorithm is proposed in [77], where an eRRH is excluded from the cluster when the eRRH contributes less to the cluster, resulting reduced backhaul burden satisfying the individual capacity constraint. On the contrary, an eRRH that contributes more to the cluster is involved to guarantee required Quality-of-Service (QoS). The involvement of an eRRH depends on backhaul capacity, requested data, and cached data.

B. Caching: Fronthaul Latency and Edge Latency

For arbitrary caching in a F-RAN, *pre-fetching* and *delivery* are main two phases. When the content popularity remains constant, pre-fetching operates in large-time scale with multiple transmission intervals. However, the delivery phase mainly operates on each one of the transmission intervals. This phase depends on message content that is already cached during the pre-fetching phase. The fronthaul-aware approaches are broadly divided into hard- [111], [113], [114] and soft-transfer [78] mode. In the hard-transfer mode, the total content that is not present in the local server was transferred from BBU to eRRHs. On the other hand, a *quantized* version of baseband signal encoded at BBU is delivered to eRRHs in soft-transfer mode. Previous work considers pre-coding

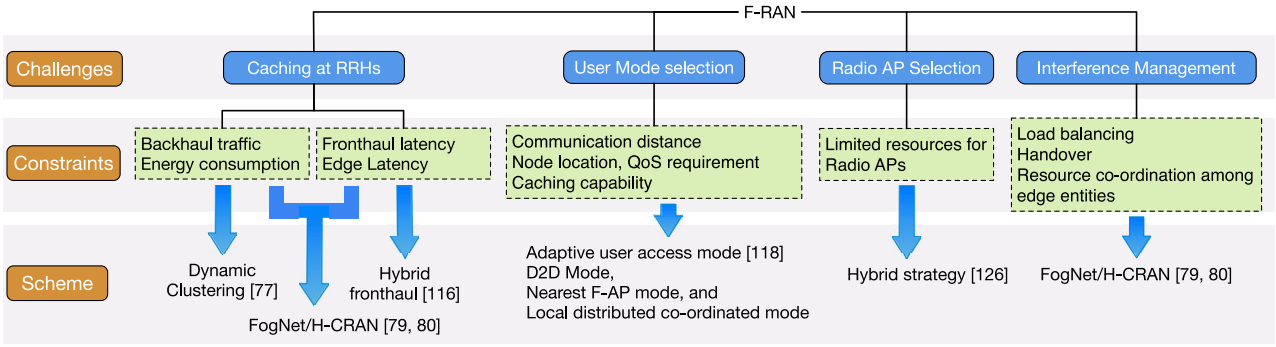

Fig. 15. An illustration of challenges, constraints, and proposed schemes in F-RAN.

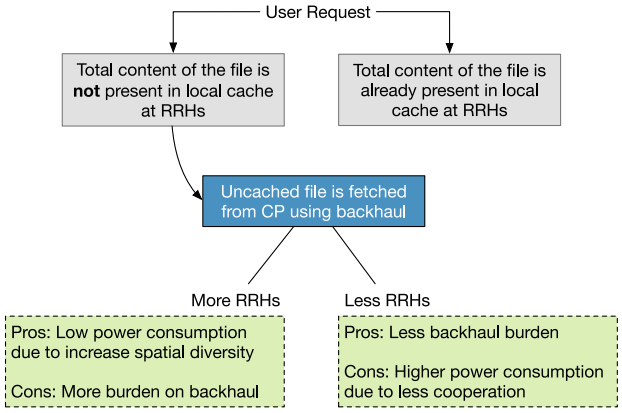

Fig. 16. File caching at eRRHs in F-RAN. A tradeoff between burden on backhaul and energy consumption is observed in caching at eRRHs in F-RAN.

optimization and fronthaul transmission strategy under different constraint, e.g., minimum delivery rate [78], the degree of freedom [115], or compound network energy cost [111], [114]. However, the delivery latency is not properly considered in [78], [111], [115], and [114].

To this end, a scheme is addressed in [116] that considers delivery latency including *fronthaul* latency between BBU and eRRHs and *edge* latency between eRRHs and EUs. The delivery phase is designed under a hybrid fronthauling mode in which fronthaul link carries both hard and soft information about the uncached file in eRRHs. Basically, this work aims to minimize the service latency with per-eRRH fronthaul capacity constraint and power limitation. An iterative algorithm is further derived based on concave-convex procedure. From the numerical results, it is argued that soft-transfer fronthauling performs better than hard-transfer when fronthaul latency dominates in overall latency. In other cases, hard-transfer mode provides a low-latency solution. This is in contrast to [78] which shows that soft-transfer fronthauling always is an effective way in almost all cases.

C. User Access Mode Selection

To reduce the burden of fronthaul between eRRHs and BBU pools in a Cloud, D2D communication, where the users do not necessarily to be connected to cloud computing unit for data transfer, is one of the promising approaches. D2D communication that efficiently utilizes the available radio resources supports proximity-based services and content sharing. The Fog computing-enabled Access Points (F-APs) are equipped with a local cache and computing capability for local processing. Apart from traditional underlay and overlay cellular network [21], [22], inter-tier and intra-tier interference limit the overall performance of both D2D and F-APs-assisted users. A novel mobile task offloading framework in D2D network underlying cellular network is suggested in [117]. The main aim is to obtain energy-efficient task execution while minimizing time-average energy consumption for the execution of all users. To this end, an online task scheduling algorithm based on Lyapunov optimization method is proposed. Various frame lengths, e.g., short, medium, and long frame on the task scheduling are considered depending on available D2D links and number of tasks to be executed. This work also jointly considers user incentive and collaboration over the network.

The stochastic geometry tool is used to derive coverage probability and ergodic rate for both F-AP-assisted and D2D users. Three user access modes are discussed as follows [118]:

- *D2D Mode:* Assuming the desired user supports D2D mode if the requested content can be obtained from another D2D-enabled user within a pre-defined distance and if Signal-to-Interference Ratio (SIR) of the user pair is above than the SIR threshold, then D2D mode is selected.
- *Nearest F-AP mode:* If the desired user does not support the D2D mode, or requested content is not present in the nearby D2D pair, or SIR is less than the predefined threshold, then desired user tries to access nearby F-APs. Then, this F-AP can support the user while satisfying SIR threshold.
- *Local distributed co-ordinated mode:* In this mode, the desired user is associated with multiple F-APs in a user-centric cluster. F-RAN adjusts the cluster radius to satisfy the content request and QoS in terms of SIR.

Afterward, an adaptive user access mode is proposed in [118] by taking the advantages of above three user modes based on *communication distance*, *node location*, *QoS requirement*, and *caching capability*. The tradeoff that depends on cache size, user density, and SIR constraint is observed.

D. Radio Access Point Selection

Fog computing allows ubiquitous connectivity and provides pooled resources in proximity to the mobile devices. Radio APs in a cluster of distributed intelligence are assumed to be equipped with computing resources. These clusters often form a local distributed computing platform. As local computing does not necessarily require to support large geographical area as in C-RAN, the computing can be satisfied by either one cloud-enabled base station or federating several base station.

However, managing resources for a set of radio APs forming a local *'mini-cloud'* [37] is one of the main research challenges. Computation offloading to a local cloud is well-discussed in [119] and [120], however, local cloud communication was not considered in detail. A centralized approach for resource management has several limitations such as delay due to the handling of a large number of users and APs, whereas decentralized schemes cannot guarantee user's Quality-of-Experience (QoE). Several scenarios, such as single-user single-cloud [121], multi-user single-cloud [122], multi-user multi-cloud [123], resource allocation inside of cluster [124], and multi-user [125] are studied. In the mult-cloud scenario, every cloud is considered as a set of cloud-enabled APs clustered together. Most recently, a strategy is proposed in [126] where the APs dynamically form a computation cluster by choosing particular APs that can be included in computation offloading. In this network, it is assumed that any mobile device is associated with only one Cloud-enabled SC. In addition, each mobile device sends the offloading computation request with maximum *allowable* response time. In proposed strategy, each serving SC builds its own computation cluster based on link quality and capacity in decentralized phase. These SCs can also get the link quality and capacity information from neighboring SCs. In a centralized phase, each serving SC sends its load distribution to central management unit for global load balancing. It is shown that this hybrid scheme [126] that combines the benefits of both centralized and decentralized approaches significantly outperforms both decentralized and centralized approach in terms of user's QoE requirement and latency, respectively. However, energy consumption issue is not discussed, which is an important metric for radio AP section in F-RAN.

E. A Hybrid Architecture: FogNet/H-CRAN

Although F-RAN is being extensively discussed, it has several problems as follows: *Reliability* of data delivery is one of the primary concern. In addition, lack of resource coordination among edge entities create interference resulting performance degradation. Although it is expected that FogNet can extend the storage capacity, however, without a proper exchange of control signaling, it is not always possible for one EU to know whether the data is present at another device in FogNet or not. A proper handover is necessary to guarantee other wireless links can operate in their own optimal way. To deal with above issues, a new RAN architecture is proposed in [79] and [80] to combine the advantages of both H-CRAN and FogNet. This proposed mechanism switches between H-CRAN and FogNet to fully utilize the network and radio resource from the viewpoint of cache, mobility, and access control.

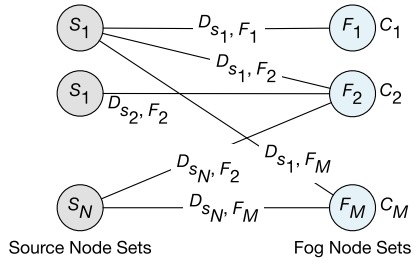

Fig. 17. A complete bipartite graph to represent the relationship between N number of source nodes and M number of fog nodes.

V. Service Allocation and Resource Management Issues in Fog Computing

In fog computing paradigm, it becomes an important issue how to decide which task or service should be processed in the fog layer or be forwarded to the cloud. Thus, service allocation is crucial to support low latency, mobile, and real-time applications. At the same time, to handle the processing of large amount of geographically distributed node, the computing and storage resources of fog layer must be efficiently utilized. In the following, several service allocation and resource management issues are discussed.

A. Latency

1) Selection of Fog Nodes (With Computing and Communication Latency Constraint): It is true that both *computing and communication latency* change according to the service request from EUs and available resources in a fog layer [99]. In some scenario, a *heavy* workload that leads to high computing delay in a particular fog node can be distributed to another fog node. However, the communication delay might be higher than previous. To design a low latency architecture, a mathematical model is suggested as follows. A group of EUs in the same area is considered as a source node $S_i, \{i = 1, 2, \ldots, N\}$ and a fog node is denoted as $F_j, \{j = i = 1, 2, \ldots, M\}$, where N and M are the total number of source nodes and fog nodes in the network. The relationship between a fog node and a source node is represented as a complete bipartite graph as shown in Fig. 17, where each source node S_i is connected to a fog node F_j. A source node S_i produces a workload w_k with an arrival rate that follows Poisson process. It is also assumed that a source node can avail any one of fog nodes in the network.

Assuming the linear function for the computing time as in (1), to select the target fog node for a given workload, an objective function is established as [99]:

$$\arg\min_{w} \quad \frac{\sum_{r=1}^{\Omega} w_{\mathrm{r}}}{\sum_{k=1}^{\Psi} w_k} \tag{19a}$$

$$\text{subject to:} \quad T_{\text{total}}(w_a) \leq \text{threshold}; \tag{19b}$$

$$0 \leq n_j \leq n_j^{\max}; \tag{19c}$$

$$n_j^{\max}, D_{S_i,F_j}, w_k > 0, \tag{19d}$$

where Ω and Ψ are the total number of blocked attempts to select the fog node for a workload and total number of workload in the network. The above objective function aims to find

minimum number of *blocking probability* which is expressed as the ratio between number of rejected workload and total number of all workload in the network. The constraint (19b) considers that communication and computation delay must be under a *pre-defined* total latency threshold of the requested workload.

To find out target fog node among all available fog nodes, following policies are suggested as [99]:

- *Random Policy:* In this policy, a fog node is randomly and uniformly selected without considering any other factors.
- *Lowest Latency Policy:* A fog node that provides minimum total latency at the current state of the system is selected for the given workload.
- *Maximum Available Capacity Policy:* This policy selects a fog node which has maximum remaining resource.

It is shown that both *random policy* and *maximum available capacity policy* provide almost same blocking probability. Although, the random policy aims to balance workload through randomness, lowest latency policy outperforms other two policies. Since a fog node finishes the workload in a shorter time in lowest latency policy compared to other two policies, this fog node can release its own resource allowing to accept other workloads, which results in lower blocking probability. This study is an early step towards the low-latency network, however, more accurate analysis with different application requirements would be an interesting extension.

2) Service Distribution Among Fog Nodes With Service Latency Constraint: Service Allocation (SA) problem in the CFC model [55], [56] refers to the challenges for EU's services with various *delay constraint and capacity*. The SA problem in [54], [56], and [127] aims to minimize *service latency* to avail resources while satisfying the service requirements. Although, several studies focus on control action required for SA, most of them consider either Cloud [127] or mobile cloud scenario [128]. Compared to the SA approach in [56] through convex optimization technique, Souza *et al.* [54] consider the *additional delay* due to both 1) SA inside fog and 2) intrinsic latency of mobile nodes [129]. The above problem is modeled as an Integer Linear Programming (ILP) model for latency optimization. To minimize total service latency, the objective of this SA problem becomes [54]:

$$\text{minimize:} \quad \sum_{i=0}^{|S|} D_i \tag{20a}$$

$$\text{subject to:} \quad \sum_{r=0}^{|R|} \sum_{k=0}^{|K_r|} Y_{i,r,k} = U_i, \forall i \in S; \tag{20b}$$

$$\sum_{k=0}^{|K_r|} \sum_{i=0}^{|S|} Y_{i,r,k} \leq K_r, \forall r \in R; \tag{20c}$$

$$\sum_{i=0}^{|S|} Y_{i,r,k} \leq 1, \forall r \in R, \forall k \in K; \tag{20d}$$

$$\sum_{r=0}^{|R|} \sum_{k=0}^{|K_r|} Y_{i,r,k} \times N_r = D_i, \tag{20e}$$

where

$$Y_{i,r,k} = \begin{cases} 1 & \text{if the } i\text{th service is allocated in the } k\text{th} \\ & \text{slot of the } r\text{th resource} \\ 0, & \text{otherwise} \end{cases},$$

and S denotes the set of services, R is the set of all available resource blocks offered by all TNs, K_r is the set of specific available resource of the rth TN, U_i is the requirement of the ith service, D_i is the total delay due to resource allocation for the ith service, and N_r is the slot allocation time for the rth resource. The constraint (20c) refers to capacity constraint to avoid the allocation of more slots than the available slots, (20d) is to prevent the usage of same slots by more than one service simultaneously. Here, the constraint (20e) considers the delay for each allocated slot of a service, where slot refers to the measurement unit to represent the resource allocation. The SA among multiple fog nodes avoids high delay to avail service from the cloud. In addition, the advantage of service distribution among fog nodes with different computing and storage capability is discussed. Nevertheless, the maximum allowable delay and impact on time overhead due to service distribution are the part of future work.

B. Energy Consumption

Fog computing is foreseen to be a viable solution towards reduced power consumption and low service latency in cloud DCs. A tradeoff between power consumption and transmission delay is investigated in [130]. Sarkar *et al.* [53] discussed the advantages of DC's power consumption and reduced service latency in fog computing compared to traditional cloud computing. The impact of CO_2 emission and cost due to operational cost and CO_2 penalty cost is discussed from a fog computing point of view.

1) Selection of Energy Efficient Applications: As suggested in [104], nDCs can be useful for fog computing, authors investigated what type of applications are more energy efficient while running in nDCs rather than centralized DCs. The applications running on nDCs are broadly categorized as a) the applications for which the data source are primarily in terminal node with either static, or b) dynamic content, c) applications for which the source of data is not created in a user terminal, rather need to be pre-downloaded to nDCs from other sources. Jalali *et al.* [104] suggested system design factors like [104] a) type of access attached to nDCs, b) ratio of active to idle time of nDCs (i.e., ϕ in (17)), and c) type of applications with factors like number of downloads, number of data pre-loading, and number of updated from the origin. They argued that the hop count between user and content has less impact compared to above factors. Basically, the main idea is to use the nDC for the applications that generate and distribute a large amount of data in EU premises. We present a logical relationship among the resource management and service allocation schemes for fog computing in Fig. 18.

C. Resource Sharing

1) A Game Theoretic Approach for Resource Sharing: Most recently, Zhang *et al.* [70] considered the scenario with

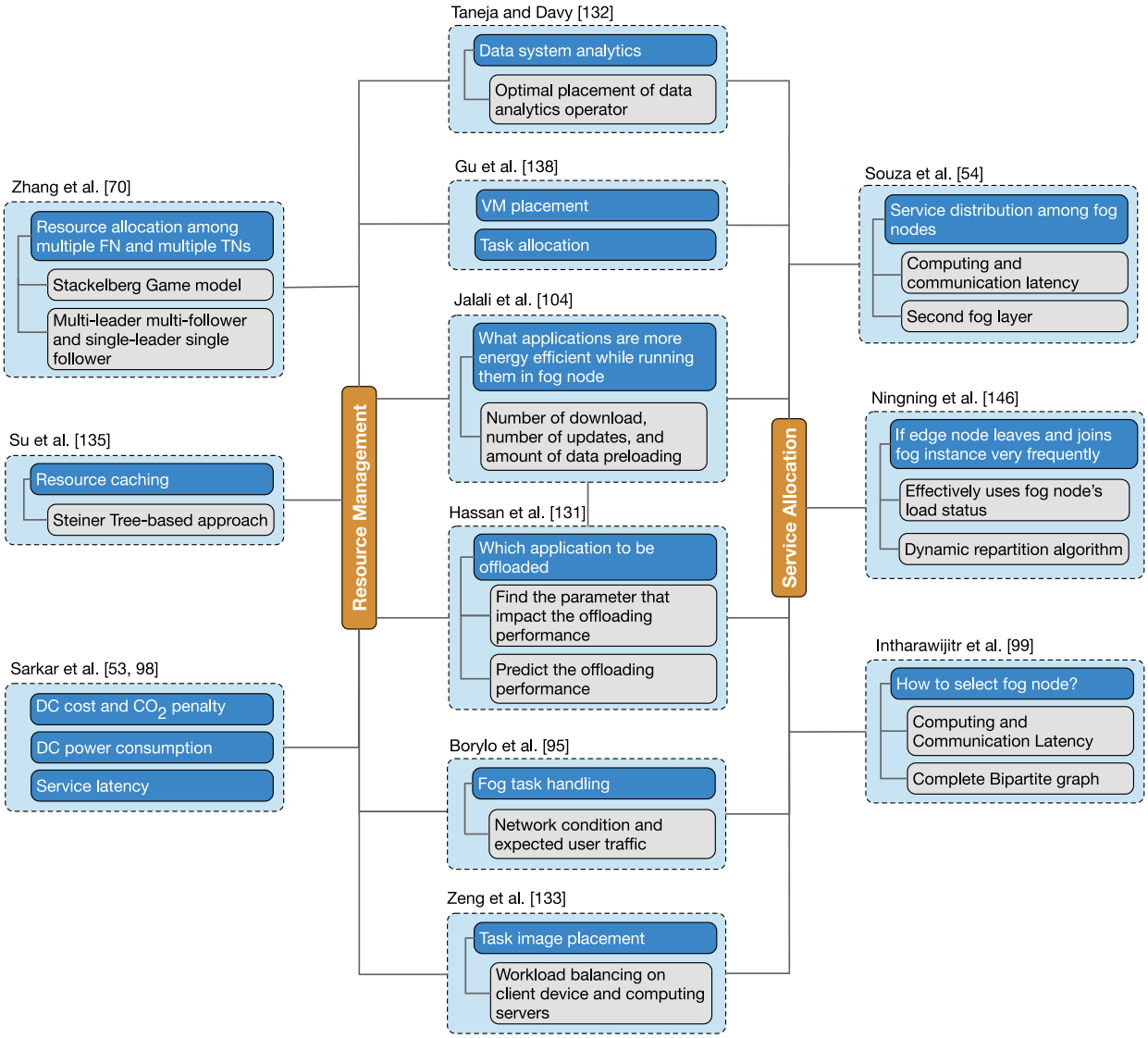

Fig. 18. Logical relationship among various schemes for resource management and service allocation in fog computing.

multiple FNs, multiple MDCOs, and multiple DSSs. In this network model, computing resource allocation to DSS in both physical and virtual network is one of the important issues. Due to the heterogeneous architecture [70] in multi-tier data center architecture with FNs, MDCOs, and DSSs, it is hard to satisfy the requirements for all FNs, MDCOs, and DSSs simultaneously. Thus, a sequential decision-making process was proposed based on game theory as shown in Fig. 19.

First, FN sets the price of its unused resources and sells the resource block to MDCOs based on the requirements of DSSs. After receiving the declared price of FNs, each one of MDCOs determines whether to execute the service requested by DSS in its own data centre or offload the service to an FN. Afterward, each MDCO determines the amount of resource block that it would like to purchase from particular MDCOs by observing the price set by all MDCOs. Then, MDCOs determine the service price to increase their revenue. In a virtualized network, each DSS determines the amount of resource block that it would like to purchase from a particular MDCO by observing the price set by all MDCOs. Since all FNs, MDCOs, and DSSs can autonomously make the decision in a distributed way, a data center pricing model as a hierarchical game approach was suggested. In that game model, the interaction between all FNs and MDCOs are considered as multi-leader and multi-follower Stackelberg game. In addition, as the interaction between MDCO and DSS is independent of each other, single-leader and single-follower Stackelberg game model was considered. The proposed resource management can able to satisfy the required utilities of all FNs, MDCOs, and DSSs. Finally, Nash equilibrium solutions were reached for all FNs.

2) Computation Offloading: Computation offloading is one of the features of fog computing. However, it is not always suitable to offload computation due to several issues. Hassan *et al.* [131] suggested finding the parameters that impact the offloading performance regarding bandwidth usage, latency, and server-side memory utilization. Afterward, the offloading performance is predicted accurately. Authors

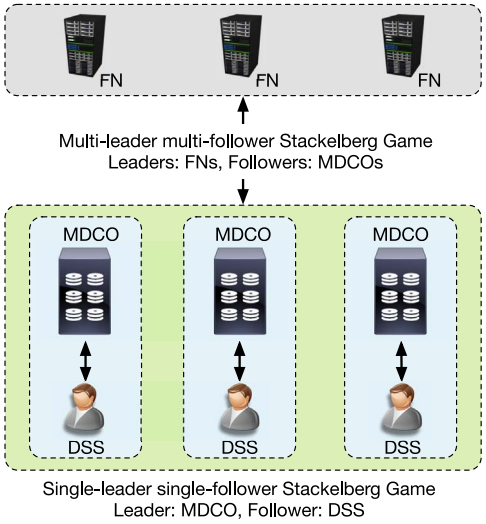

Fig. 19. Stackelberg game model is proposed for multiple FNs, multiple MDCOs, and multiple DSSs. The interactions between all FNs and MDCOs are considered as multi-leader and multi-follower Stackelberg game, whereas the interaction between MDCO and DSS is considered as single-leader and single-follower Stackelberg game model.

suggested a way to decide which tasks to be offloaded. In addition, placement of data analytics is also an important design factor. Taneja and Davy [132] briefly introduced where to place the data analytics operator that results in maximum efficiency and reduced network traffic.

3) Others: Furthermore, a resource management and task image placement are considered in fog computing-based software-defined embedded system [133]. In the software-defined embedded system, task image generally resides in storage server while the task execution is performed on either embedded device or a computation server. The objective functions and constraints are formulated as MINLP problem. Basically, workload balancing on the client device and computing servers is studied towards a computation-efficient approach.

D. Caching

1) Proactive Caching of Computing Tasks: Proactive computing [102] has been studied in wireless content caching [25], [26] where the content of the upcoming tasks are predicted and prefetched during the current task computation. On the other hand, proactive caching of computing tasks, another alternative to content caching, prefetches the computing results during an off-peak interval, thus reduces the backhaul burden. Therefore, in fog networks, proactive caching of computing tasks that reduce the burden on access link by the computing results in advance without any prefetched content of the tasks is an important issue. Elbamby *et al.* [27] suggested a proactive caching of popular and cacheable computing tasks considering both computing and storage resources to minimize service latency. Due to large network size, the EUs are clustered into disjoint groups based on distance-based Gaussian similarity and task-popularity-based similarity. The cloudlet aims to minimize the latency for the tasks requested by the EUs by proactively caching the computation results. Only processing delay occurs for the cached tasks in the cloudlet. Initially, the cache is empty, then it gets occupied as the computing results are stored. Furthermore, the cloudlet replaces computing results for the less popular tasks by the high popular task. Besides, a game theoretic approach [134] is suggested in [27] between EUs and cloudlets to minimize the overall latency in cache-enabled for networks.

2) Resource Caching: The fog clusters are mainly responsible for providing service to the EUs benefiting the following advantages as a) user terminals do not need to store many resources as they can retrieve resource from fog clusters, b) need not share resource among other terminal nodes resulting reduced energy consumption, and c) low latency while retrieving resources. Therefore, to achieve this goal, all the resources need to be shared or cached in fog clusters to provide service to user terminals. An important question is how to share or cached resource among fog cluster with minimum cost [135].

To improve the performance of fog cluster layers, Su *et al.* [135] suggested an optimization process as follows. First, they assume that each fog cluster is composed by movie server, Web server, file server, and the game server, which are randomly connected to form a topology. As data transmission between any two connected servers results in cost, resource sharing among servers is one of the major issues with a constraint of connection cost. They model this scenario by a graph $G = \{N, E\}$ with N set of nodes representing servers with E set of edges that represent the connection between servers. A weight e_i is given for ith edge to represent connection cost between fog cluster servers. To find another graph $G^* = \{P, E^*\}$ with the condition of minimizing total weight of E^*. The optimization problem becomes

$$\text{minimize} \sum_{i \in E^*} e_i$$

$$\text{subject to} \begin{cases} P \subset N \\ G^* \text{is connected} \end{cases}$$

To solve the above problem, Steiner tree [136], [137] was used to minimize total edge weight of the connected graph. Stainer tree aims to find the shortest interconnection for a given set of nodes. The basic difference between minimum spanning tree and Steiner tree-based approaches is that additional intermediate node and edges may be added in Steiner tree to find an optimum connection with reduced cost compared to the minimum spanning tree. The proposed approach outperforms traditional shortest path scheme to find an optimum way for shared/cached resources in the fog servers of the fog Cluster.

E. Placement of Virtual Machine (VM)

As fog computing emerges as a promising solution, Gu *et al.* [138] introduced a framework towards cost effective Medical CPS (MCPS). It is suggested that apart from communication service to the medical devices, Base Station (BS) can act as fog computing device that provide computation and storage to host Virtual Machine Device (VMD) applications in Virtual Machines (VMs). Therefore, it aims to reduce

the data uploading and computational overhead by processing the medical data with the corresponding VMs applications in VM residing at BSs instead of sending to centralized DC. The computational overhead can be reduced if all VMs are placed in associated BSs. However, as VM requires a certain amount of resources to ensure QoS, therefore placing all VMs is not possible due to limited resource of BSs. To solve these issues, they first formulated BS association, task distribution, and VM placement problem as a Mixed-Integer Non-Linear Programming (MINLP). Afterward, a Mixed-Integer Linear Programming (MILP) problem was suggested. The main objective is to minimize overall unit cost for deploying Fog Computing-based MCPS (FC-MCPS) in a given BS infrastructure to guarantee the required QoS. Finally, a low-complexity two-phase Linear Programming (LP)-based heuristic algorithm is proposed to address computational complexity.

VMs live migration: In the context of DC virtualization, VM live mitigation becomes an essential function to replace the running VMs seamlessly across the physical devices [139], in order to save the energy consumption. In live mitigation, although the image of the physical memory is transferred to the new destination, the previous VM still continues the previous process for the completion. Several technologies for VM live migration, e.g., XenMotion [139] and VMware [63]. VMotion are widely used to find out the when a VM should be migrate. However, the bandwidth issues are not considered at the time and place of VM mitigation. The channel optimization for intra-data center is discussed in [140]. Afterward, the bandwidth optimization on wireless channel is presented in [141], which primary focuses on Pre-Copy Migration (PeCM), however, can be easily applied to other VM mitigation techniques such Stop-and-Copy Migration (SaCM), Post-Copy Migration (PoCM) and HyBrid Migration (HyBM). The proposed optimization approach considered the constrains on the total migration time, downtime and overall available bandwidth.

F. Uncertainty of Position and Availability of Fog Nodes

The terminal nodes in fog computing change their states such as active or inactive, joining or leaving any fog computing instances more frequently compared to the traditional cloud computing [142]–[144]. Therefore load balancing becomes very difficult due to communication overhead and dynamic resource allocation. In addition, latency is also an important factor in many delay-sensitive applications. Based on the control mode, load balancing in traditional cloud computing is broadly divided into centralized [143], [144] and distributed policy [145]. The central control unit manages the load of each servers of the cloud computing, however, results bottleneck in performance. Although the distributed schemes have flexibility, they suffer from very complex architecture.

1) Dynamic Load Balancing Mechanism: The existing methods [143]–[145] have the following shortcomings as fault tolerance, load forecasting, and system hierarchy, therefore, cannot be directly applied to the dynamic architecture of fog computing. Recently, Ningning *et al.* [146] introduced a dynamic repartition algorithm for fog computing with the use of cloud atomization process. They first combined the graph theory and the characteristics of fog computing. Then, the physical nodes in a graph are mapped into a virtual machine node in that graph. Afterward graph partitioning of virtual machine nodes is performed. This results in low control overhead when the resource or task changes with time. Particularly, in static load balancing algorithm, the system recalculates the task load for each node. However, the dynamic repartition algorithm [146] effectively uses the node's load status and balances the system without moving a large amount of load when any node exits from or joins to the system. It is observed that this dynamic approach performs better than static load balancing algorithm for fog computing architecture.

2) Online Fog Network Formation: Although many prior works assume that the formation of fog network is known, the joining and leaving of fog node are very uncertain due to several reasons. In addition, the presence of neighboring node is highly uncertain. An online approach [100] is designed for fog formation considering the uncertainty of fog nodes and minimizing the computational latency. Based on the online k-secretary framework [147], in exploration stage, a fog node decides how to offload the computation tasks between other fog nodes and the cloud server. Basically, the fog node selects the most suitable neighbouring fog node to construct the network. Afterward, an optimization algorithm is formulated to minimize the overall computation delay of all fog nodes by suitably selecting the set of fog nodes. Particularly, the proposed framework [100] allows to select the set of fog nodes with relaxing the constraint of prior position and availability of fog nodes, thereafter to distribute the computation tasks across fog and cloud.

G. Flexible Infrastructure for Telecom Operator

To support a wide range of user requirements in terms of latency, resiliency, bandwidth, and security for 5G [148], a distributed architecture is required in fog networks. Among many challenges in fog networks such as fog-cloud interaction, edge and D2D communication, integration with biodata and analytics framework, wireless networking is an important issue to place fog nodes at the edge of the networks. Recently, TelcoFog architecture [149] is introduced for the telecom operator to provide network functions virtualization (NFV), MEC, and IoT services benefiting the dynamic deployment of distributed low-latency services. Basically, the proposed architecture has the following three blocks as: *fog node* is seamlessly integrated with the networking infrastructure, *controller* assures the service, and *services* are able to run on the top of network architecture. The controller also supports the dynamic deployment of the generic visualized functions. The YANG [150] data modelling language is used to model the TelcoFog services. Basically, the TelcoFog nodes enable a scalable framework for the distributed and programmable fog technologies.

Fig. 20. Typical applications supported by fog computing.

VI. Applications of Fog Computing

Fog computing is foreseen to be a viable solution to support the tasks with bandwidth, latency, and reliability constraint. Basically, fog computing has a huge potential to fulfill the diverse requirements in various applications ranging from smart cities [40], [151]–[156], transportation [157]–[159], surveillance (e.g., face recognition [160]) and healthcare [161]–[163] to large-scale industries [164]–[167]. Many of these applications are also predicted by OpenFog consortium [17] (shown in Fig. 20). As a real-world example, augmented and virtual reality (AR/VR), where virtual and physical information merge seamlessly, over wireless can be accomplished by the advantages of edge/fog/cloud computing [168]. Note that a prototype, called remote reality prototype Mirrorsys [169], requires 100-200 Mbps for a one-way immersive response. While most of the solutions for AR/VR allocate computing and communication resources in a centralized manner, these resources must be moved towards the edge of a network to support at least < 13 ms latency to enable real-time visual and auditory information for the VR users [168]. Several tasks[1] such as two-way telepresence (5 to 25 Mbps), 3D model and data visualization (2 to 20 Mbps), Current-Get 360 video (4K) (10 to 50 Mbps), and next-gen 360 video (90+ Frames per second, HDR) can be supported by fog nodes. Since fog computing supports a wide range of applications, some fog node provides large storage to host data and applications, routing and switching, and others services such as configuration, QoS, security. Some of the typical applications and how fog computing can be applied to overcome the major limitations in these applications are summarized in Table VI. The details are presented as follows.

A. Smart Cities

Some case studies are presented with the characteristics of fog computing in smart cities [40]. For example, in smart agriculture [155], a part of smart cities, fog computing plays an important role to monitor plant growth and climate condition with smart sensor nodes. In such cases, location-awareness [156], [170] and almost real-time response are important factors by incorporating the context information. Furthermore, smart-transportation [175] and smart waste management [170] are suggested in [40] as a case study in fog-enabled smart cities. In these scenarios, the data collection and aggregation model in fog computing are highlighted. In addition, water management, greenhouse gases control, and retail automation are some of the examples where fog computing can be widely adopted in the smart cities.

To enable deployment and execution of multiple location-aware and low-latency applications leveraging CPS as smart city facility, a fog computing-based infrastructure is proposed in [151]. This infrastructure acts as a tunneling and forwarding agent enabling inter-object communication at data-link and network layer. To implement Sensing and Architecture-as-a-Service (SAssS) [176] approach, Stack4Things middleware [152] is proposed. This middleware provides complex event processing for performing basic reconfigurable functionality in the smart cities. A case study is demonstrated where the vehicles interact with city-level smart objects to support real-time and geo-localized services. However, more scalable deployment is required to enable interaction between these smart objects.

As an application of smart cities, in smart urban surveillance, fog computing can be used for critical and dynamic tasks. In [154], a fog computing-based system tracks the vehicles in real-time and enables multi-target tracking using single tracking algorithm. A homogeneous system architecture to bridge the devices in highly heterogeneous network edge is also suggested as a future work. Furthermore, it is suggested that due to the social relationship among EUs, specific service requests can be easily handled in SIoT, where cyber counterparts SVO can reside in the cloud. A four-level SIoT cloud-based architecture is proposed in [153] where virtualization layer can be implemented in both traditional cloud and fog architecture to ensure trustworthiness among the edge resources.

B. Smart Energy Management

As power grid becomes very popular with microgrids which are comprised of the widely distributed generator, power supply, and loads, energy management is essential to balance power generation and consumption for the residential, industrial, and commercial domain. Several organizations such as U.S. Department Of Energy (DOE) [177] aims to reduce the national energy consumption in both residential and commercial buildings. Al Faruque and Vatanparvar [171] considered Home Energy Management (HEM), as an application of sophisticated and low-cost energy management system, and found the main research challenges while implementing energy management platform as:

- Performance, interoperability, and interactivity among heterogeneous devices in energy management platform.
- Ability to customize the services, adaptability, and scalability of the energy management platform for various types of buildings, homes, and applications.
- Cost of implementing the energy management platform, hardware, and software stack.

Cloud computing, a centralized approach for computing, provides the customers with infrastructure, platform, software, and sensor networks. However, as services [178]–[180] with

[1]https://www.qualcomm.com/news/onq/2017/02/01/vr-and-ar-are-pushing-limits-connectivity-5g-our-rescue

TABLE VI
TYPICAL APPLICATIONS OF FOG COMPUTING PARADIGM

Applications	Characteristics and main limitations	How fog computing helps to overcome these limitations
Smart cities	• Multiple location-aware and low latency applications, where most of the logic is executed on smart objects itself.	• Stack4Things middleware is proposed to implement SAssS [151], [152] approach.
	• To handle critical and dynamic tasks during real-time vehicle tracking.	• Smart urban surveillance based on fog computing [154]. • Enables multi-target tracking using single tracking algorithm.
	• Due to social relationship among end devices specific service requests can be searched in a very effective manner.	• Cyber counterparts SVO can reside in the cloud. Virtualization layer can be implemented in both traditional cloud and fog architecture [153].
	• Smart waste management that includes different process, e.g., waste collection, transport, processing, and monitoring of waste materials.	• Fog computing can reduce the data communication by opportunistically uploading the data to the cloud [170].
	• Plant growth and climate condition in a timely and location-based manner.	• Intelligent reconfiguration to save energy by eliminating unnecessary sensing and communication [156], [170].
Smart energy management	• Interoperability and interactivity among heterogeneous devices in energy management platform. • Ability to customize the services, adaptability, and scalability of the energy management platform for various types of buildings, homes, and applications.	• Energy management-as-a service [171]. • Fog computing platform provides the flexibility, interoperability, connectivity, data privacy, and real-time features required for energy management.
Industrial wireless sensor networks	• Industrial systems consists of Cloud, DCs, edge, and TN domain which pose different software environment.	• A framework, called continuous computing [167] allows seamless computing environment across multi-domain applications.
	• High data collection and processing capability • Low power consumption, and • Real-time application.	• An energy-efficient and delay-aware WCS is proposed [164] .
	• Handles different protocol and data format from highly dissimilar data sources.	• Adaptive operations Platform is suggested to improve operational efficiency [165].
	• How to decide which data should go to cloud or be processed in fog layer	• A scheme with an aim to reduce traffic overhead leveraging storage and computing services toward edge devices [166].
Remote gaming	• Off-load the on-demand content.	• A gaming platform is suggested for high extensibility, portability, and reconfigurability [172], [173].
Healthcare	• Complexity and delay occur in emergency notification.	• Due to the low latency, context-awareness, and localized nature, fog computing is used [161]. • Information related to emergency events are synchronized with the cloud for further analysis and early warning.
	• To reduce latency in response time.	• A fog computing-based brain-computer interaction [163] • Data is preprocessed in smartphone acting as fog nodes
	• Where to host VMD instances to bind a clinical algorithm and a set of the related physical devices.	• A fog computing-based approach, FC-MCPS [162], is proposed towards efficient resource allocation management. • Base Station association, task distribution, and VM placement have jointly considered for resource optimization.
Intelligent transport system	• To support high computation and communication for vehicular applications.	• Vehicular fog computing (VFC) [157] is proposed. • It employs vehicle-as-infrastructure to extend communication and computation resources. • The relationship among connectivity, communication and mobility of vehicles.
	• Vehicles and human enter or leave from vehicular network very often, access control becomes an important issue in context-aware services [174] to protect the privacy of EU.	• Attribute-based access control scheme [158] is addressed.
	• Application development is challenging due to distributed nature of fog computing.	• Basic requirements of application model and programming abstraction [159] are discussed.
	• Privacy and • Fairness challenges.	• Vehicular fog node provides the local service by optimizing several parameter of vehicular networks. • Several privacy measurements are suggested [43]. • Fairness challenges in vehicular crowdsensing applications are introduced.

reliability and performance agreements are essential for the energy management platform, increasing the number of devices in the energy management system are not always suitable for some delay-sensitive devices with timing requirements [181], [182]. In contrast, fog computing provides the capability of pre-processing the data while meeting the low latency requirements [12]. However, fog computing must satisfy the above-mentioned requirements in energy management system.

The devices in hardware architecture of energy management platform are mainly categorized as connecting devices, gateway, sensors, actuators, and computing devices. The gateway device maintains the compatibility between multiple devices. Various kind of sensors gathers analog signal generated by the environment. Actuators, which can be optimized for energy consumption, may be either locally or globally controlled. The computing device implemented in the sophisticated controller can store, process, and analyze data. The opensource

and user-configurable routers [183] can be used as computing node to configure the controller easily. Different routing and discovery algorithm for collecting data can be programmed on the low-powered wireless sensors, e.g., TelosB module with TinyOS [184].

The energy management, which was implemented on fog computing platform [171] can be used for any buildings and various domains of operation, e.g., home or microgrid. The energy management may have various purposes like 1) monitoring and metering the power consumption of each device, e.g., home power consumption; 2) managing the energy consumption by controlling the devices efficiently, e.g., intelligent lighting, Electric Vehicle (EV) charger [185], Heating, Ventilation, and Air Conditioning (HVAC) management. The energy management platform is a system of systems. Thus, the hardware, software, and communication architectures should be defined and integrated properly to design the platform for these systems.

C. Remote Gaming: Gaming-As-a-Service

As mobile gaming is one of the most demanding applications, remote gaming becomes very popular. In a remote gaming system [186]–[189], EUs do not need to download or install any games to tablets, phone, or other devices, instead they can play the game from a server over the Internet. In a remote cloud gaming, the game is being executed and processed in the cloud and users can receive the video and audio streaming while they can control the game from their end. In this way, a user can save a lot of energy of hand-held devices. From a game company point of view, some attractive benefits are as follows: 1) *gaming-as-a-service* [186] can protect the games from privacy, 2) upgrading of games becomes very easy, 3) use more homogeneous platform for developing of games, 4) cheating can be prevented due to storing all game state at the server, and 5) publishing game sessions becomes easier to attract new users.

However, the hand-held devices in EUs suffer from limited energy. The approach to offload the on-demand is one of the possible ways to use mobile as a thin-client and handle all remaining computing at the server side. Mobile computing either aim to save energy or platform the task that cannot be performed solely by the mobile devices. Nevertheless, moving the computation over the network also consumes energy [187]. A cloudlet architecture was proposed in which the cloud already contains a base image. A device only transfers the delta image (e.g., Apache Web server) and its configuration. By employing cloudlet [29] for gaming, Kämäräinen *et al.* [190] proposed an approach where mobile device acts a thin-client and starts gaming service on-demand. GamingAnywhere [172], [173] was adopted as a gaming platform that is designed for high extensibility, portability and reconfigurability is a promising open-source project to implement remote gaming on the mobile devices. The GA server and GA client are shown in Fig. 21. Kämäräinen *et al.* [190] compared various delay-sensitive game running in a cloudlet [29] versus Amazon EC2 data centers [191]. In addition, to understand the energy efficiency of GA at client-side, 'thin-client' approach for remote gaming was proposed in [192]. A significant energy saving was obtained in the tested devices. It was also observed that the energy saving could be increased in multi-player games. Along with commercialization of GA, server-side virtualization while optimizing the energy consumption can be the possible future direction.

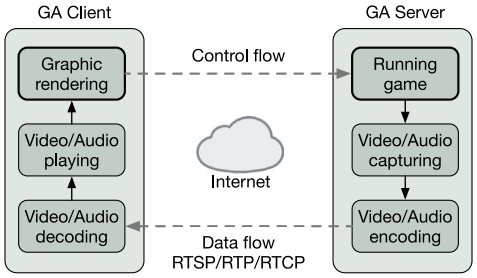

Fig. 21. GamingAnywhere (GA) [172], [173] consists of GA server and GA client. GA server obtains the audio and video frames generated by the selected games at the server side and GA client has the following tasks: 1) decoding the received data, 2) rendering the video/audio frames, and 3) dealing with the players input events [192].

D. Industrial Wireless Sensor Networks

Large-scale industrial applications are generally associated with robots, fixtures, machine tools, workpieces, and chemical reaction, etc. With Industry 4.0 [193]–[197], smart factories with Industrial Wireless Sensor Networks (IWSNs) play a significant role in factory automation, production, manufacturing, and safety. A massive amount of data are generated from manufacturing, production process, and real-time monitoring. Data gathered by widely distributed industrial devices are normally sent to the centralized data centers for data processing and storage. However, volume and velocity of IoT data exchange cost communication resources due to the increased number of connected things. As a result, bottleneck arises regarding *data processing*, *latency*, and *traffic overhead*.

In fog-based IWSNs [166], the fog nodes that are physically close to sensor node directly communicate with gateway nodes. Traditional applications are provided through the Internet in cloud-based IWSNs [198], whereas, the fog-based network aims to significantly reduce the feedback latency and computation overhead. As fog computing serves as a virtual layer between traditional cloud computing and edge devices [98], [165], *how to decide which data should go to the cloud or be processed in fog layer* is one of the basic challenges. From an industrial application point of view, the main requirements in the fog nodes are as follows [164]:

- Real-time response,
- Simultaneous data collection from large heterogeneous sources, and
- Low power consumption.

To fulfill some of the above requirements, Suto *et al.* [164] suggested a Wireless Computing System (WCS), which was originally used for rack architecture in DCNs [199], as a fog computing node in IWSNs. As many servers each with two 60 GHz can be accommodated in limited space, WCS [164] has very high data collecting and processing capabilities. For

low power consumption requirement, an energy-efficient and delay-aware WCS was proposed to control sleep scheduling and network connectivity with an acceptable delay constraint. This scheme carefully selects the number of servers in the sleep-state and degree of servers in the active-state to effectively handle power consumption and delay constraint compared to other approaches.

Also, with the benefits of fog computing, the main research challenges arise how to handle different protocols and data formats from highly dissimilar data sources in fog layer. Note that the traditional approaches for data acquisition and normalization, which were implemented in the cloud, are not well-suited in resource-constrained fog layer. Furthermore, to estimate the malfunction and failure of any production process in an industry, a failure model was introduced based on a set of assumptions regarding the behavior of a respective product. Although the failure model that requires additional information about the operational context for more accurate estimation of failure is under the operational administration of each product. Hence, this model is not available to the equipment manufacturer. In addition, these failure model were initially designed for a large number of products with fewer types, thus, are not well-suited for the small-scale production of fully customizable items.

Adaptive Operations Platform (AOP) [165] is suggested to improve the effectiveness of the failure model. Basically, AOP is based on the service provided by following layers: *fog infrastructure* with fog resources and end-to-end services, and *operational support system* that provides the asset management, production, and manufacturing support function. Furthermore, the key functional elements in the AOP are discussed as follows:

- *Model Building (MB):* To combine the static information of a failure model of the equipment in the industrial area with dynamic data collection, a fused model can be build up to express a refined model for appropriate industrial infrastructure to better fit in an operational context.
- *Rule Mapper (RM):* It maps the fused model to a set of traffic handling rules.
- *Rule Deployer (RD):* This element computes the deployment plan to apply traffic handling rule on appropriate elements.

To show the effectiveness of the proposed approach [165], a Cisco ISR 819 router [200], [201] that supports Data in Motion (DMo) to manage IoT data is used. From testbed, it is argued that operational efficiency is significantly improved with better utilization of communication and computing resources by introducing the above approach in the industrial applications.

The large companies, even medium-sized enterprises, run backend applications in edge domain that consists of general purpose hardware, e.g., industrial computer, workstations, and servers. Each domain like cloud, edge or device plays a major role in using different software environment, resulting in the static assignment of system functionality. Continuous computing is considered in [167] to implement identical software environment across different computer domain. This approach supports design, development, and deploying software over multi-domain applications. A reference model is suggested to support the functionality of proposed continuous computing. However, this platform should be able to support real-time and the device capability in the industrial applications.

E. Healthcare

As a healthcare application, Zao *et al.* [163] proposed a fog computing-based brain-computer interaction where the Electroencephalogram (EEG) data is preprocessed in the smartphones to reduce the service latency. Particularly, an emergency alert in health and environment requires very low latency and fast processing of context-aware data [202]. Nevertheless, cloud computing platform is very useful in emergency applications [203], [204] to store and process a large amount of data with flexible and scalable virtual servers and virtual networks. On the other hand, due to the low latency, context-awareness, and localized nature, fog computing is expected to support emergency and healthcare related services. With these inherent nature of fog computing, Aazam and Huh [161] introduced a smartphone-based service, called an Emergency Help Alert Mobile Cloud (E-HAMC) system where fog computing acts as an offloading layer and sends the location and other emergency details to the concerned department. Meanwhile, the information related to emergency also synchronized with a cloud for further analysis and future early notification to the users as well as service provider. Basically, this scheme tries to overcome the complexity and delay in emergency notification [202]. This architecture also handles prank emergency notification while avoiding location spoofing. From the preliminary experiments, it is observed that the proposed scheme obtains a notable gain regarding quick and easy-access to the victims compared to direct communication with the cloud. Furthermore, a self-awareness health-care system is suggested in [205] where MIST [206] computing and fog computing principles are applied for data collection and processing of human body parameters.

Generally, CPS [207] enables seamless integration between the physical component and cyber services such as networking and computation. As an emerging application of CPS, Medical CPS (MCPS) [162], [208], [209] aims to improve the life quality of human being. Basically, MCPS integrates the medical devices with software applications. MCPS provide high-quality healthcare services [210] by enabling automatic controlling of the delivery devices based on the data from the monitoring devices. In MCPS, low-capability sensors and actuators often require interaction with external software for full functionality. For example, Apple watch and iPhone sense the data and send to a control platform to store and analyze the data. The VMD [162] instances bind a clinical algorithm and a set of related physical devices. To host these VMD applications, cloud data center is one of an ideal place. However, data from the physical devices must be transferred in real-time and analyzed by the VMD application to provide fast and accurate feedback. As a result, with the growing number of physical devices, MCPS suffers from long-delay and an unstable link between the cloud data center and medical devices. Gu *et al.* [138] suggested that the VMD application can be hosted in the network edge to reduce the infrastructure

burden. A fog computing-based approach, called as FC-MCPS, is proposed towards efficient resource allocation management. Base station association, task distribution, and VM placement have jointly considered while investigating the resource allocation problem with minimum cost to guarantee the required QoS in FC-MCPS.

F. Intelligent Transportation System

As an example of Vehicular Ad-hoc Networks (VANETs) applications, Intelligent Transportation System (ITS) [211] is one of the places where fog computing would be very useful. With the emergence of cloud computing, vehicular networks have gained a significant attention to ensure safety, traffic efficiency, and public convenience. Application architecture that uses transportation and vehicular devices exploits the cloud computing [212]–[216], where developed applications are executed in cloud and data are collected from the vehicular network.

However, the large cloud-to-EU delay is one major limitation for the real-time vehicular application. For example, in autonomous driving, one of the ITS applications, the vehicle needs to control the speed, which depends on real-time feedback from road-side infrastructure, traffic flow, and passengers activities. Thus, cloud computing, although provides a huge data storage and computing, is not always a solution for many time-critical applications. Fog computing is well suited in VANET [217] because of its geo-distribution, real-time load balancing, and local decision-making mechanism. Vehicles, roadside units, and other transportation infrastructure play an important role in computation process. It is true that the development of fog computing in VANET applications is very challenging than cloud computing-based VANET applications due to high-mobility and latency-sensitive applications. An adaptive resource management approach is proposed in [218] to support real-time vehicular TCP/IP-based fog platform. Furthermore, Giang *et al.* [159] discuss the requirements such as reusable, scalable, context-aware [156], [174], and high-level abstraction in applications model and programming abstractions. It is suggested that this platform must support low-latency, decentralized computation, and data-flow communications. Furthermore, it is very common phenomena that both vehicles and EUs are leaving and entering to the fog cluster arbitrarily. To support the context-aware services [174] in ITS, personal information including location is stored and processed in fog as well as the cloud server. In that sense, access-control becomes an important issue in the fog computing-based ITS. An attribute-based access control scheme [158] is suggested for protection of privacy.

Heterogeneity, dynamic nature of smart transportation and large-scale co-ordination are some of the main research challenges in fog computing-based transportation systems. Since smart transportation system consists of various devices with different computing capability, heterogeneity becomes an essential issue from fog computing perspective. Also, due to dynamic nature of smart transportation system, there is always a trade-off between the adaptation of system-wide implementation and application layer. Therefore, flexible execution platform is required to balance between difficulty to implement and burden on the developer. As all entities are spread over a large geographical area, smart transportation systems need to handle a huge number of connected devices. As a result, since intelligent transportation systems do not inherit routing layer as in the traditional WSN, large-scale co-ordination becomes a challenging task.

Recently, a Fog-based Vehicular Crowdsensing (FVCS) [43] architecture is proposed to integrate the mobile crowdsensing with fog computing. Several applications such as parking navigation, road surface monitoring, and traffic collision monitoring are recommended. Moreover, the security and privacy preserving measurements are suggested to protect the location and identity information for the collected data from the surroundings. Also, fairness challenges are highlighted. A novel idea, Vehicular Fog Computing (VFC) [157], is proposed that utilizes the near-use edge devices to make the better utilization of vehicular computation and communication resources. It also uses the under-utilized computation and communication resources of both moving and parked vehicular infrastructure. Apart from the fog computing architecture in [12] where vehicular and mobile devices belong to the TNs, this proposed VFC architecture is different in the context that these vehicular and mobile devices are also a part of fog node. VFC also depends on near-location instead of sending them to a remote server. In this way, the critical situations can be supported in a better way compared to Vehicular Cloud Computing (VCC) [212], [213]. It is suggested that estimating the mobility model of the vehicles will help to better utilize these resources. Also, security and incentive issues for the EUs and the vehicular devices [43] are the part of future directions.

VII. Open Research Challenges and Future Directions

Finally, some of the open research challenges are summarized in Table VII. In this section, we endeavor to present open research challenges as follows.

A. Application Offloading

By computation and storage offloading facilities, fog computing benefits the applications that are not effective while running them on resource constraint EU's devices. However, in certain instances, application offloading is not always efficient regarding the delay, bandwidth, and energy consumption due to lack of available resources in fog layer [104], [131]. Thus, before offloading any application, it is equally important to find the parameters that have an impact on the offloading performance. In addition, it would be useful if we can accurately predict the performance of the offloaded applications. If the predicted performance does not gain any significant advantages, then offloading to the fog is not efficient.

Placement of Real-time Analytics: A real-time analytics plays an important role for application offloading in fog computing. In [219], a scalable, energy-efficient, generic and extensible component-based distributed data analytics platform is presented for a typical distributed data analytics in fog-cloud computing. The analytics should be able to find

TABLE VII
SUMMARY OF STATE-OF-THE-ART AND RESEARCH CHALLENGES

Challenges	Description	Focus/Objective	Contribution	Research opportunities
Load Balancing [146] or Service Allocation [54]	• Edge nodes changes their state, join, and leave any fog computing instances very frequently.	• Maps physical node graph model into a virtual machine node model [146] • Cloud atomization process on physical node.	• Graph repartitioning of virtual machine nodes.	To further improve the performance of dynamic load balancing algorithms.
	• To support different delay-constraint requirement as well as low capacity of service requests.	• To minimize service latency to avail resources	• A QoS-aware service allocation problem is formulated as integer optimization problem [54].	• The impact of time overhead by service allocation. • Maximum allowable delay to avail resources.
BS association, task distribution, and VM deployment [138]	• Data uploading and computational overhead are reduced by processing the data with the corresponding VMS applications in VM residing at BSs instead of sending to centralized data center	• To minimize overall unit cost for deploying FC-MCPS in a given BS infrastructure to guarantee the required QoS.	• A MINLP, afterward, MILP problem was suggested. • A low-complexity two-phase LP- based heuristic algorithm is proposed to address computational complexity.	• Other optimization approaches with more parameters like various time-critical service requests could be interesting.
Energy consumption [104]	• Some applications that generate and distribute data continuously in TN premises can save energy using nano DCs (nDCs).	• To find what type of applications are more energy efficient while running in nDCs rather than centralized DCs rather using centralized DCs. • To derive an comprehensive energy consumption model for content distribution.	• Found the parameters like number of downloads, number of updates, and the amount of data pre-loading for the applications that can use nDCs.	• Further identify the applications that can save energy while running from nDCs rather than centralized DCS
DC's power consumption	• Carbon emission of traditional DC is significant due to round the clock operation and huge data processing.	• How to design an architecture to reduce carbon emission.	• A fog computing framework is proposed towards practical implementation [53]	• Fog resource management
Cache or resource sharing [70], [135]	• All the resources in fog cluster need to be shared to provide access to user terminals	To find optimum way to share resource with minimum cost [135].	Steiner tree [136], [137]-based approach to analyze resource sharing in fog clusters	• To consider the service latency comprise of communication and computation latency.
	• Data center virtualization with multi-tier data center architecture is considered.	To satisfy the requirements simultaneously for all FNs, MDCOs, and DSSs with heterogeneous architecture of the data center network.	• Stackelberg game model is proposed to find optimal resource sharing among multiple FNs, multiple MDCOs, and multiple DSSs [70].	• More parameters should be considered while designing the cost between leader and follower.
	• Resource management and task image placement	• Task image generally resides in storage server while the task execution is performed on either embedded device or a computation server.	• Fog computing-based software-defined embedded system [133] • Workload balancing on client device and computing servers is studied towards a computation-efficient approach.	• Memory management issues
Fog-related task handling	• Computation offloading is not always efficient.	• To find the parameters that impact the offloading performance in terms of bandwidth usage, latency, and server-side memory utilization. • How to predict the offloading performance in an accurate way.	• Suggested a way to decide which tasks to be offloaded [131].	• To consider service availability and security-related issues.
	• Due to fog resource outage, some tasks cannot be supported in fog node	• To direct latency-aware fog tasks to cloud in order to maintain required QoS.	• The SDN controller communicates with both fog and cloud to handle delay-aware tasks [95].	• Optimal DC placement
	• Where to place the data analytics operator that results maximum efficiency and reduced network traffic.	• Placement of data analytics to optimize resource allocation and task distribution.	• Service placement algorithm [132]	• Real-world performance evaluation

what applications can be offloaded to satisfy the QoS requirement. It also can help to decide which data should be sent to either cloud or fog layer. To find optimum placement for real-time analytics [132] that results in maximum efficiency and reduced network traffic is another important issue. Finally, a dynamically learning algorithm [219] can be useful to enable fine-grained control of sensing and data offloading frequency.

B. Resource Management

Compared to cloud computing, the fog computing does not enough computing and storage resources. Thus, efficient

resource allocation is an important research issue in fog computing.

Mobility: Due to dynamic nature, the EU and fog nodes leave (join) from (to) a fog layer arbitrarily. Thus, the dynamic load balancing and task assignment to mobile end device become very difficult due to the scalability issue in fog layer. Therefore, how to balance the load towards cost-effective concerning delay, power consumption, and bandwidth resource distribution is one of the open research directions. Since fog computing is localized, *understanding the mobility pattern* of end devices may be helpful for task assignment and resource management in fog computing.

Virtualization: Small-scale DCs at the network edge can offload the computing task and storage requirements, after that, reduce the traffic over the network and infrastructure cost compared to large remote DCs. Although, these edge DCs benefits in many aspects such as better QoS and low communication delay, virtualization is one of the key approaches to support VDCs with different service requirements and objectives.

C. Fog Radio Access Networks (F-RANs)

In F-RAN, eRRHs are equipped with a local cache to minimize the backhaul (and fronthaul) burden due to the highly popular on-demand data request. By this way, F-RAN is an attractive solution to reduce traffic congestion compared to traditional C-RAN. Nevertheless, file caching plays an essential role in F-RAN.

Number of eRRH for Uncached File Sharing: The eRRH caches the popular file until its memory is filled. However, the uncached file still creates a burden on the backhaul. In this case, the number of eRRHs is an important factor to be considered. If more eRRHs are involved for uncached file sharing, then the power consumption becomes less, however, the backhaul burden will increase. On the other hand, if fewer eRRHs are involved, energy consumption will increase due to less cooperation. Therefore, the tradeoff between power consumption and backhaul burden must be considered.

Caching Size: The cache size of eRRHs is also an important factor since the cached file affects the latency and backhaul burden in F-RAN.

User Mode Selection: The users in F-AP-assisted and D2D mode either underlaying or overlaying cellular networks could use the following ways of communication and data sharing. If the desired user's content can be obtained from another D2D-enabled user above a predefined SIR value, then the D2D mode is selected. In another way, if the desired user cannot support D2D mode or the content is not available in the nearby D2D pair within a threshold limit, then the desired user aims to access the nearby F-AP. Apart from one single F-AP, multiple F-APs can collaboratively assist the desired users. Thus, communication distance, node location, and caching capability influence the user mode selection. Apart from these, the interference that arises due to the coexistence of F-APs, D2D pair, and the cellular network cannot be ignored in F-RANs.

Latency: Although the latency is significantly reduced in F-RAN compared to C-RAN, the fronthaul and backhaul latency between BBU and eRRHs, as well as eRRH and EU, are significant research challenges in F-RAN.

D. Heterogeneity

As the bottom-most layer in fog computing architecture consists of various end devices such as a smartphone, smartwatch, virtual sensor node, intelligent devices including autonomous car, smart home devices, the heterogeneity issue arises regarding data collection, data format, and data processing capability. Nevertheless, fog node in fog cluster comprises by routers, switches, gateway and other devices with different computing and communication facilities. Thus heterogeneity becomes an important design factor in fog computing architecture. Handling of different data formats and various communication protocols for managing semi- or unstructured data becomes major issues.

E. Standardization

The working group of OpenFog consortium [17] is working towards standardization of fog computing architecture in terms of communication; security, testbeds, manageability, and software Infrastructure. Standardization is required so that *various IoT systems can securely interact each other and cloud service.* This also enables computing, networking, and storage across multiple edges, fog, and end devices.

F. Incentives

To obtain high QoS, such as low service latency, sufficient storage, high bandwidth, and fast computation for the service sometimes require a cost in terms of money. Nevertheless, *pay-as-you-go* seems to be an essential aspect in terms of incentives for the users who share their computing and storage resources [117]. Proper cost management and how to properly bid the computing and storage *price* towards a 'trustful' collaboration are worthwhile to be studied in fog computing.

G. SDN-Based Fog Computing

With the benefits of fog computing, then communication and computation burden on the SDN controller can be reduced. On the other hand, SDN benefits from improved scalability, resource management, and reduced configuration cost in fog computing. Nevertheless, in SDN-based fog computing, several research issues arise as follows:

Local Coordinator: In SDN-based fog architecture, a local coordinator [88] is required to handle the dynamic and time-varying service requirement. This coordinator will decide which task should be processed in the fog or be sent to the cloud. Moreover, the optimal placement of coordinator is also an important design issue. Therefore, fog-related task handling becomes one of the major tasks for the local coordinator.

Controller Design: The fog-SDN controllers need to cooperate among themselves to efficiently handle the limited fog resources [87], [88]. The learning pattern form the previous statistics help to predict the data traffic, thereafter, improve the resource allocation. Finally, it is equally important to implement the controller's intelligence in the different level of the application programming interfaces (API).

TABLE VIII
LIST OF ABBREVIATION

AP	Access Point
BBU	Base-Band signal processing Unit
BS	Base Station
C-RANs	Cloud Radio Access Networks
CFC	Combined Fog-Cloud
CP	Centralized Processor
CPS	Cyber-Physical Systems
D2D	Device-to-Device communication
DC	Data Center
DSS	Data Service Subscriber
DTN	Delay-Tollerant Networks
EE	Energy Efficiency
eRRHs	Enhanced Remote Radio Heads
EU	End User
F-AP	Fog computing-enabled Access Point
F-RANs	Fog Radio Access Networks
FogNet	Fog Network
H-CRAN	Heterogeneous-CRAN
HetNet	Heterogeneous Network
HPN	High Power Node
IaaS	Infrastructure-as-a-Service
IoT	Internet of Things
ITS	Intelligent Transport System
IWSN	Industrial Wireless Sensor Network
LAA	Licensed Assisted Access
LTE	Long Term Evolution
LTE-A	LTE-Advanced
MDCO	Massive DC Operator
MDS	Maximum Distance Separable
MEs	Micro Engines
mmWave	Millimetre Wave
nDC	Nano-Data Center
PaaS	Platform-as-a-Service
QoE	Quality-of-Experience
QoS	Quality-of-Service
RA	Reference Architecture
RAC	Radio Access Controller
RAT	Radio Access Technologies
RRHs	Remote Radio Heads
RU	Radio Unit
RWOs	Real-World Objects
SaaS	Software-as-a-Service
SC	Small Cell
SDN	Software-Defined Networking
SE	Spectral Efficiency
SIoT	Social Internet of Things
SIR	Signal-to-Interference Ratio
SVO	Social Virtual Object
TN	Terminal Node
UMTS	Universal Mobile Telecommunications System
VANET	Vehicular Ad-hoc Networks
VCC	Vehicular Cloud Computing
VFC	Vehicular Fog Computing
VM	Virtual Machine
VMD	Virtual Machine Device
WCS	Wireless Computing System

Optimizing the Reconfiguration Cost: An orchestration mechanism is required to disseminate data forwarding rule change and service hosting for the services implemented at VMs in the SDN controller in fog computing [91], [220]. This orchestration should handle the service instantiation, replication, and mitigation. Although, fog-SDN controllers can automatically update the service hosting and data forwarding rules, reconfiguration to data flow rules and service instantiation, replication, and mitigation affect Quality-of-Experience (QoE) and service latency [220], [221]. Therefore, optimizing the reconfiguration cost becomes one of the major issues in SDN-based fog computing.

Implementation of Test-bed: Due to the centralized nature of SDN, it is challenging to implement fog computing at the edge switches. A few steps are already taken [93], [95], [96], however, still in infant stage. How to implement fog computing in off-the-shelf network switches and to manage resources at network switches are still an active research topic. The flowtable management and non-overlapping OpenFlow rules [222] in fog-enabled SDN switches are still open issues.

VIII. CONCLUSION

Fog computing has the potential to support delay sensitive service requests with low traffic congestion, low energy consumption and minimum bandwidth with an aim to reduce the burden on cloud data centers. Fog computing, which is not a replacement for cloud computing, extends the computation, communication, and storage facilities from cloud to edge of the networks. Although, fog computing is a viable solution towards sustainable development of IoT market, many unsolved issues still exist. This survey presents an overview of various architectures discussed in the literature and identifies the key research challenges. Since fog computing is foreseen as a next computing paradigm, it can be applied to a wide range of network applications. Finally, this survey presents some of the open research challenges and basic design principles outlined above. In summery, the aim of this survey is to provide fundamentals and network applications of fog computing that help us to understand previous research contributions and future research direction to solve different challenges in fog computing.

APPENDIX
ABBREVIATIONS

See Table VIII.

REFERENCES

[1] *IDC, Worldwide Internet of Things Forecast Update 2015–2019*, document #US40983216, IDC, Framingham, MA, USA, Feb. 2016.
[2] *IDC Forecasts Worldwide Shipments of Wearables to Surpass 200 Million in 2019, Driven by Strong Smartwatch Growth and the Emergence of Smarter Watches*, document #prUS41100116, IDC, Framingham, MA, USA, Mar. 2016.
[3] *IDC, Worldwide Internet of Things Forecast Update, 2017–2021*, document #US43304017, IDC, Framingham, MA, USA, Feb. 2018.
[4] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, May 2014.
[5] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun.*, Dalian, China, Sep. 2008, pp. 5–13.
[6] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. 5th Int. Joint Conf. INC IMS IDC*, Seoul, South Korea, Aug. 2009, pp. 44–51.
[7] *Data Center Companies*. Accessed: Mar. 20, 2017. [Online]. Available: https://www.datacenters.com/directory/companies

[8] *NRDC, Americas Data Centers Consuming and Wasting Growing Amounts Energy*, Feb. 2015. [Online]. Available: https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy

[9] *Apple Hits 100% Renewable Energy in Its Data Centers*. Accessed: Sep. 20, 2017. [Online]. Available: http://www.datacenterknowledge.com/archives/2013/03/22/apple-hits-100-renewable-energy-in-its-data-centers/

[10] *Apples Next U.S. Data Center Will Be Built in Iowa*. Accessed: Sep. 20, 2017. [Online]. Available: https://www.apple.com/newsroom/2017/08/apples-next-us-data-center-will-be-built-in-iowa/

[11] *The Carbon Footprint of the Internet*. Accessed: Sep. 12, 2017. [Online]. Available: https://www.custommade.com/blog/carbon-footprint-of-internet/

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

[13] *Cisco Delivers Vision of Fog Computing to Accelerate Value From Billions of Connected Devices*, Cisco, San Jose, CA, USA, Jan. 2014. [Online]. Available: https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1334100

[14] M. Aazam and E.-N. Huh, "Fog Computing: The cloud-IoT/IoE middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May/Jun. 2016.

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

[16] *Definition of Fog Computing*. Accessed: May 12, 2017. [Online]. Available: https://www.openfogconsortium.org/resources/#definition-of-fog-computing

[17] *OpenFog Consortium*. Accessed: Mar. 20, 2017. [Online]. Available: https://www.openfogconsortium.org

[18] *OpenFog Reference Architecture*. Accessed: Mar. 20, 2017. [Online]. Available: https://www.openfogconsortium.org/ra/

[19] *Intelligent Edge Intel*. Accessed: Feb. 20, 2017. [Online]. Available: https://itpeernetwork.intel.com/extending-intelligence-to-the-edge/

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

[21] H. ElSawy, E. Hossain, and M.-S. Alouini, "Analytical modeling of mode selection and power control for underlay D2D communication in cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4147–4161, Nov. 2014.

[22] D. Feng *et al.*, "Device-to-device communications underlaying cellular networks," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3541–3551, Aug. 2013.

[23] M. Wildemeersch, T. Q. S. Quek, M. Kountouris, A. Rabbachin, and C. H. Slump, "Successive interference cancellation in heterogeneous networks," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4440–4453, Dec. 2014.

[24] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "RAT selection games in HetNets," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 998–1006.

[25] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. IEEE 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, Aug. 2014, pp. 945–949.

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

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

[28] A. Bader, H. Ghazzai, A. Kadri, and M.-S. Alouini, "Front-end intelligence for large-scale application-oriented Internet-of-Things," *IEEE Access*, vol. 4, pp. 3257–3272, 2016.

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

[30] G. A. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, and J. Root, "Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments," in *Proc. ACM Companion 36th Int. Conf. Softw. Eng.*, 2014, pp. 412–415.

[31] A. M. M. Ali, N. M. Ahmad, and A. H. M. Amin, "Cloudlet-based cyber foraging framework for distributed video surveillance provisioning," in *Proc. IEEE 4th World Congr. Inf. Commun. Technol. (WICT)*, Dec. 2014, pp. 199–204.

[32] C. Mouradian *et al.*, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2017.

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

[34] 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.

[35] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," Sophia Antipolis, France, ETSI, White Paper, vol. 11, Sep. 2015.

[36] *Multi-Access Edge Computing*. Accessed: Jan. 20, 2018. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[37] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Singapore: Springer, 2018, pp. 103–130.

[38] K. P. Saharan and A. Kumar, "Fog in comparison to cloud: A survey," *Int. J. Comput. Appl.*, vol. 122, no. 3, pp. 10–12, Jul. 2015.

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

[40] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Survey*, vol. 50, no. 3, pp. 1–43, 2017.

[41] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017.

[42] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Warsaw, Poland, Sep. 2014, pp. 1–8.

[43] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 146–152, Jun. 2017.

[44] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 601–628, 1st Quart., 2018.

[45] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[46] M. Mukherjee *et al.*, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017.

[47] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Australasian Telecommun. Netw. Appl. Conf. (ATNAC)*, Southbank, VIC, Australia, Nov. 2014, pp. 117–122.

[48] P. Kumar, N. Zaidi, and T. Choudhury, "Fog computing: Common security issues and proposed countermeasures," in *Proc. Int. Conf. Syst. Model. Advancement Res. Trends (SMART)*, Nov. 2016, pp. 311–315.

[49] Q. Yaseen, F. Albalas, Y. Jararwah, and M. Al-Ayyoub, "Leveraging fog computing and software defined systems for selective forwarding attacks detection in mobile wireless sensor networks," *Trans. Emerg. Telecommun. Technol.*, pp. 1–13, May 2017.

[50] Q. Yaseen, F. AlBalas, Y. Jararweh, and M. Al-Ayyoub, "A fog computing based system for selective forwarding detection in mobile wireless sensor networks," in *Proc. IEEE 1st Int. Workshops Found. Appl. Self Syst. (FAS W)*, Augsburg, Germany, Sep. 2016, pp. 256–262.

[51] S. Anwar *et al.*, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions," *MDPI Algorithms*, vol. 10, no. 2, pp. 1–24, Mar. 2017.

[52] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 739–752, Jan. 2018.

[53] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan./Mar. 2018.

[54] V. B. C. Souza *et al.*, "Handling service allocation in combined fog-cloud scenarios," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.

[55] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 120–128, Oct. 2016.

[56] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE ICC*, London, U.K., Jun. 2015, pp. 3909–3914.

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

[58] M. Chen, H. Jin, Y. Wen, and V. C. M. Leung, "Enabling technologies for future data center networking: A primer," *IEEE Netw.*, vol. 27, no. 4, pp. 8–15, Jul./Aug. 2013.

[59] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, 1st Quart., 2016.

[60] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Amsterdam, The Netherlands: Morgan Kaufmann, 2004.

[61] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[62] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. C-34, no. 10, pp. 892–901, Oct. 1985.

[63] *VMware*. Accessed: Feb. 15, 2017. [Online]. Available: http://www.vmware.com

[64] *Xen*. Accessed: Feb. 15, 2017. [Online]. Available: http://xen.org

[65] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Jan. 2009.

[66] B. Ahlgren *et al.*, "Content, connectivity, and cloud: Ingredients for the network of the future," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 62–70, Jul. 2011.

[67] S. Islam and J.-C. Grégoire, "Network edge intelligence for the emerging next-generation Internet," *Future Internet*, vol. 2, no. 4, pp. 603–623, Nov. 2010.

[68] K. Church, A. Greenberg, and J. Hamilton, "On delivering embarrassingly distributed cloud services," in *Proc. ACM HotNets*, Oct. 2008, pp. 55–60.

[69] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the Internet with nano data centers," in *Proc. ACM 5th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Rome, Italy, Dec. 2009, pp. 37–48.

[70] H. Zhang *et al.*, "Fog computing in multi-tier data center networks: A hierarchical game approach," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[71] C. Guo *et al.*, "SecondNet: A data center network virtualization architecture with bandwidth guarantees," in *Proc. 6th Int. COnf. (Co-NEXT)*, Philadelphia, PA, USA, Nov./Dec. 2010, Art. no. 15.

[72] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, "A hierarchical game framework for resource management in fog computing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 52–57, Aug. 2017.

[73] H. Zhang *et al.*, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[74] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, "Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems," *J. Commun. Netw.*, vol. 18, no. 2, pp. 135–149, Apr. 2016.

[75] S.-H. Park, O. Simeone, O. Sahin, and S. S. Shitz, "Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 69–79, Nov. 2014.

[76] "White paper: China mobile. C-RAN: The road towards green RAN," China Mobile Res. Inst., Brussels, Belgium, Rep. 2.5, Oct. 2011. [Online]. Available: http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper_v2_5_EN(1).pdf

[77] D. Chen, S. Schedler, and V. Kuehn, "Backhaul traffic balancing and dynamic content-centric clustering for the downlink of fog radio access network," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Edinburgh, U.K., Jul. 2016, pp. 1–5.

[78] S.-H. Park, O. Simeone, and S. S. Shitz. (Jan. 2016). *Joint Optimization of Cloud and Edge Processing for Fog Radio Access Networks*. [Online]. Available: https://arxiv.org/pdf/1601.02460.pdf

[79] 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.

[80] 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.

[81] M. Peng, K. Zhang, J. Jiang, J. Wang, and W. Wang, "Energy-efficient resource assignment and power allocation in heterogeneous cloud radio access networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 5275–5287, Nov. 2015.

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

[83] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2713–2737, 4th Quart., 2016.

[84] "OpenFlow switch specification (version 1.5.1)," Open Netw. Found., Menlo Park, CA, USA, Rep. ONF TS-025, Mar. 2015. [Online]. Available: http://www.opennetworking.org

[85] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 351–362, Aug. 2010.

[86] Y. Li *et al.*, "Multiple mobile data offloading through delay tolerant networks," in *Proc. 6th ACM Workshop Challenged Netw.*, Sep. 2011, pp. 43–48.

[87] K. Liang, L. Zhao, X. Chu, and H.-H. Chen, "An integrated architecture for software defined and virtualized radio access networks with fog computing," *IEEE Netw.*, vol. 31, no. 1, pp. 80–87, Jan./Feb. 2017.

[88] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep. 2017.

[89] L. Huang *et al.*, "Software-defined QoS provisioning for fog computing advanced wireless sensor networks," in *Proc. IEEE SENSORS*, Orlando, FL, USA, Oct./Nov. 2016, pp. 1–3.

[90] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2017.

[91] 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. Manag. (IM)*, May 2015, pp. 1202–1207.

[92] 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.

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

[94] *MQTT—Message Queuing Telemetry Transport*. Accessed: Mar. 20, 2017. [Online]. Available: http://mqtt.org

[95] 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 ICC*, May 2016, pp. 1–7.

[96] Y. Xu, V. Mahendran, and S. Radhakrishnan, "SDN docker: Enabling application auto-docking/undocking in edge switch," in *Proc. IEEE INFOCOM Workshops*, Apr. 2016, pp. 864–869.

[97] L. Zhang *et al.*, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2710–2721, Dec. 2013.

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

[99] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–4.

[100] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[101] D. P. Bertsekas and R. G. Gallager, *Data Networks*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall Int., 1992.

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

[103] A. Feldmann *et al.*, "Energy trade-offs among content delivery architectures," in *Proc. CTTE*, Ghent, Belgium, Jun. 2010, pp. 1–6.

[104] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[105] A. Vishwanath *et al.*, "Energy consumption of interactive cloud-based document processing applications," in *Proc. IEEE ICC*, Jun. 2013, pp. 4212–4216.

[106] F. Jalali *et al.*, "Energy consumption of photo sharing in online social networks," in *Proc. IEEE/ACM 14th Int. Symp. Cluster Cloud Grid Comput.*, May 2014, pp. 604–611.

[107] J. Qiao *et al.*, "Enabling device-to-device communications in millimeter-wave 5G cellular networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 209–215, Jan. 2015.

[108] A. Samukic, "UMTS universal mobile telecommunications system: Development of standards for the third generation," *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1099–1104, Nov. 1998.

[109] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, New York, NY, USA, 1999, pp. 126–134.

[110] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[111] M. Tao, E. Chen, H. Zhou and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, Sep. 2016.

[112] Y. Uğur, Z. H. Awan, and A. Sezgin, "Cloud radio access networks with coded caching," in *Proc. 20th Int. ITG Workshop Smart Antennas (WSA)*, Munich, Germany, Mar. 2016, pp. 1–6.

[113] X. Peng, J.-C. Shen, J. Zhang, and K. B. Letaief, "Backhaul-aware caching placement for wireless networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[114] D. Chen, S. Schedler, and V. Kuehn, "Backhaul traffic balancing and dynamic content-centric clustering for the downlink of fog radio access network," in *Proc. IEEE 17th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Edinburgh, U.K., Jul. 2016, pp. 1–5.

[115] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 809–813.

[116] S.-H. Park, O. Simeone, and S. Shamai, "Joint cloud and edge processing for latency minimization in fog radio access networks," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2016, pp. 1–5.

[117] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D Fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[118] S. Yan, M. Peng, and W. Wang, "User access mode selection in fog computing based radio access networks," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[119] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Jun. 2010, pp. 49–62.

[120] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 945–953.

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

[122] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Darmstadt, Germany, Jun. 2013, pp. 26–30.

[123] S. Sardellitti, S. Barbarossa, and G. Scutari, "Distributed mobile cloud computing: Joint optimization of radio and computational resources," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, USA, Dec. 2014, pp. 1505–1510.

[124] J. Oueis, E. C. Strinati, and S. Barbarossa, "Small cell clustering for efficient distributed cloud computing," in *Proc. IEEE 25th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Washington, DC, USA, Sep. 2014, pp. 1474–1479.

[125] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE 81st VTC Spring*, Glasgow, U.K., May 2015, pp. 1–6.

[126] J. Oueis, E. C. Strinati, and S. Barbarossa, "Distributed mobile cloud computing: A multi-user clustering solution," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[127] O. Rogers and D. Cliff, "A financial brokerage model for cloud computing," *J. Cloud Comput. Adv. Syst. Appl.*, vol. 1, no. 1, pp. 1–12, 2012.

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

[129] X. Wang, J. Wang, X. Wang, and X. Chen, "Energy and delay tradeoff for application offloading in mobile cloud computing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 858–867, Jun. 2017.

[130] 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, Dec. 2016.

[131] 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 (SECON Workshops)*, Seattle, WA, USA, Jun. 2015, pp. 1–6.

[132] M. Taneja and A. Davy, "Poster Abstract: Resource aware placement of data stream analytics operators on fog infrastructure for Internet of Things applications," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Washington, DC, USA, Oct. 2016, pp. 113–114.

[133] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in Fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[134] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[135] J. Su, F. Lin, X. Zhou, and X. Lu, "Steiner tree based optimal resource caching scheme in fog computing," *China Commun.*, vol. 12, no. 8, pp. 161–168, Aug. 2015.

[136] *Steiner Tree Problem*. Accessed: Mar. 20, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Steiner_tree_problem

[137] A. Zelikovsky, "A series of approximation algorithms for the acyclic directed Steiner tree problem," *Algorithmica*, vol. 18, no. 1, pp. 99–110, 1997.

[138] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108–119, Jan./Mar. 2017.

[139] C. Clark *et al.*, "Live migration of virtual machines," in *Proc. ACM 2nd Conf. Symp. Netw. Syst. Design Implement. (NSDI)*, May 2005, pp. 273–286.

[140] E. Baccarelli, D. Amendola, and N. Cordeschi, "Minimum-energy bandwidth management for QoS live migration of virtual machines," *Comput. Netw.*, vol. 93, no. 1, pp. 1–22, 2015.

[141] D. Amendola, N. Cordeschi, and E. Baccarelli, "Bandwidth management VMs live migration in wireless fog computing for 5G networks," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, Oct. 2016, pp. 21–26.

[142] F. Ye, S. Wu, Q. Huang, and X. A. Wang, "A novel QoS-aware load balancing mechanism in cloud environment," in *Proc. IEEE Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Sep. 2016, pp. 298–301.

[143] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 34–40, Sep. 2012.

[144] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 702–710.

[145] G. Vilutis, L. Daugirdas, R. Kavaliūnas, K. Šutienė, and M. Vaidelys, "Model of load balancing and scheduling in cloud computing," in *Proc. IEEE 34th Int. Conf. Inf. Technol. Interfaces*, Cavtat, Croatia, Jun. 2012, pp. 117–122.

[146] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Commun.*, vol. 13, no. 3, pp. 156–164, Mar. 2016.

[147] R. Kleinberg, "A multiple-choice secretary algorithm with applications to online auctions," in *Proc. 16th ACM SIAM Symp. Discr. Algorithms (SODA)*, Vancouver, BC, Canada, Jan. 2005, pp. 630–631.

[148] *Building Scalable, Sustainable, Smart+Connected Communities With Fog Computing*. Accessed: Mar. 12, 2017. [Online]. Available: http://blogs.cisco.com/innovation/barcelona-fog-computing-poc

[149] R. Vilalta *et al.*, "TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 36–43, Aug. 2017.

[150] B. Claise. *YANG as the Data Modeling Language in the IoT Space*. Accessed: Feb. 5, 2017. [Online]. Available: https://www.iab.org/activities/workshops/iotsi/

[151] D. Bruneo *et al.*, "Stack4Things as a fog computing platform for smart city applications," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, Apr. 2016, pp. 848–853.

[152] G. Merlino, D. Bruneo, S. Distefano, F. Longo, and A. Puliafito, "Stack4Things: Integrating IoT with openstack in a smart city context," in *Proc. Int. Conf. Smart Comput. Workshops*, Hong Kong, Nov. 2014, pp. 21–28.

[153] I. Farris *et al.*, "Social virtual objects in the edge cloud," *IEEE Cloud Comput.*, vol. 2, no. 6, pp. 20–28, Nov./Dec. 2015.

[154] N. Chen, Y. Chen, S. Song, C.-T. Huang, and X. Ye, "Poster abstract: Smart urban surveillance using fog computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 95–96.

[155] C. Perera, P. P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, and P. Christen, "Sensor discovery and configuration framework for the Internet of Things paradigm," in *Proc. IEEE World Forum Internet Things (WF IoT)*, Seoul, South Korea, Mar. 2014, pp. 94–99.

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

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

[158] S. Salonikias, I. Mavridis, and D. Gritzalis, "Access control issues in utilizing fog computing for transport infrastructure," in *Critical Information Infrastructures Security*. Cham, Switzerland: Springer Int., Oct. 2016, pp. 15–26.

[159] N. K. Giang, V. C. Leung, and R. Lea, "On developing smart transportation applications in fog computing paradigm," in *Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Nov. 2016, pp. 91–98.

[160] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1910–1920, Aug. 2017.

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

[162] I. Lee *et al.*, "Challenges and research directions in medical cyber–physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 75–90, Jan. 2012.

[163] J. K. Zao *et al.*, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. IEEE Int. Conf. Intell. Environ.*, Shanghai, China, Jun. 2014, pp. 374–377.

[164] 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.

[165] V. Gazis *et al.*, "Components of fog computing in an industrial Internet of Things context," in *Proc. IEEE 12th Annu. Int. Conf. Sens. Commun. Netw. (SECON) Workshops*, Seattle, WA, USA, Jun. 2015, pp. 1–6.

[166] M. Mukherjee, L. Shu, D. Wang, K. Li, and Y. Chen, "Poster abstract: A fog computing-based framework to reduce traffic overhead in large-scale industrial applications," in *Proc. IEEE INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–2.

[167] H. Mueller, S. V. Gogouvitis, H. Haitof, A. Seitz, and B. Bruegge, "Poster abstract: Continuous computing from cloud to edge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Washington, DC, USA, Oct. 2016, pp. 97–98.

[168] E. Baştuğ, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[169] *Mirrorsys*. Accessed: Feb. 15, 2017. [Online]. Available: http://www.huawei.com/minisite/mwc2015/en/mirrorsys.html

[170] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 1, pp. 81–93, 2014.

[171] 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, Apr. 2016.

[172] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: An open cloud gaming system," in *Proc. ACM 4th Conf. Multimedia Syst. (MMSys)*, Feb. 2013, pp. 36–47.

[173] *GamingAnywhere*. Accessed: Mar. 20, 2017. [Online]. Available: http://gaminganywhere.org/

[174] G. D. Abowd *et al.*, "Towards a better understanding of context and context-awareness," in *Proc. 1st Int. Symp. Handheld Ubiquitous Comput. (HUC)*, Karlsruhe, Germany, Sep. 1999, pp. 304–307.

[175] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the Internet of Things era," *IT Prof.*, vol. 17, no. 3, pp. 32–39, May/Jun. 2015.

[176] S. Distefano, G. Merlino, and A. Puliafito, "Sensing and actuation as a service: A new development for clouds," in *Proc. IEEE 11th Int. Symp. Netw. Comput. Appl.*, Cambridge, MA, USA, Aug. 2012, pp. 272–275.

[177] Department of Energy. (2014). *U.S. Department of Energy Strategic Plan*. Accessed: Feb. 12, 2017. [Online]. Available: http://energy.gov/sites/prod/files/2011_DOE_Strategic_Plan_.pdf

[178] L. Wang *et al.*, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 4–15, Jan. 2014.

[179] S. Patidar, D. Rane, and P. Jain, "A survey paper on cloud computing," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Jan. 2012, pp. 394–398.

[180] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 26–33, Jan. 2017.

[181] A. Copie, T.-F. Fortis, and V. I. Munteanu, "Benchmarking cloud databases for the requirements of the Internet of Things," in *Proc. ITI 35th Int. Conf. Inf. Technol. Interfaces*, Jun. 2013, pp. 77–82.

[182] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proc. IEEE*, vol. 102, no. 1, pp. 11–31, Jan. 2014.

[183] *DD-WRT: Open Source Firmware for Routers*. Accessed: Feb. 12, 2017. [Online]. Available: http://dd-wrt.com/site/index

[184] P. Levis *et al.*, *TinyOS: An Operating System for Sensor Networks* (Ambient Intelligence). Heidelberg, Germany: Springer, 2005, pp. 115–148.

[185] M. A. A. Faruque, L. Dalloro, S. Zhou, H. Ludwig, and G. Lo, "Managing residential-level EV charging using network-as-automation platform (NAP) technology," in *Proc. IEEE Int. Elect. Veh. Conf.*, Greenville, SC, USA, Mar. 2012, pp. 1–6.

[186] W. Cai, V. C. M. Leung, and M. Chen, "Next generation mobile cloud gaming," in *Proc. IEEE 7th Int. Symp. Service Oriented Syst. Eng.*, Redwood City, CA, USA, Mar. 2013, pp. 551–560.

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

[188] X. Chen, Y. Chen, M. Dong, and C. Zhang, "Demystifying energy usage in smartphones," in *Proc. ACM/EDAC/IEEE 51st Design Autom. Conf. (DAC)*, San Francisco, CA, USA, Jun. 2014, pp. 1–5.

[189] X. Chen, Y. Chen, Z. Ma, and F. C. A. Fernandes, "How is energy consumed in smartphone display applications?" in *Proc. ACM 14th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Feb. 2013, pp. 1–6.

[190] T. Kämäräinen, M. Siekkinen, Y. Xiao, and A. Ylä-Jääski, "Towards pervasive and mobile gaming with distributed cloud infrastructure," in *Proc. IEEE/ACM 13th Annu. Workshop Netw. Syst. Support Games (NetGames)*, Nagoya, Japan, Dec. 2014, pp. 1–6.

[191] *Amazon-EC2*. [Online]. Available: https://aws.amazon.com/ec2/

[192] M. Komu, R. Morabito, T. Kauppinen, J. Kjallman, and Y. Yao, "Power consumption in remote gaming: An empirical evaluation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM) Workshops*, San Francisco, CA, USA, Apr. 2016, pp. 736–741.

[193] P. C. Evans and M. Annunziata, "Industrial Internet: Pushing the boundaries of minds and machines," GE, Rep., Nov. 2012. [Online]. Available: www.ge.com/docs/chapters/Industrial_Internet.pdf

[194] G. Miragliotta, A. Perego, and A. Tumino, *Internet of Things: Smart Present or Smart Future?* Dept. Manag., Econ. Ind. Eng., Politecnico di Milano Italy, Milan, Italy, 2012.

[195] H. Sasajima, T. Ishikuma, and H. Hayashi, "Future IIoT in process automation—Latest trends of standardization in industrial automation," in *Proc. IEEE 54th Annu. Conf. Soc. Instrum. Control Eng. Japan (SICE)*, Hangzhou, China, Jul. 2015, pp. 963–967.

[196] M. R. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, "Applications of wireless sensor networks in the oil, gas and resources industries," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Perth, WA, Australia, Apr. 2010, pp. 941–948.

[197] Q. Chi, H. Yan, C. Zhang, Z. Pang, and L. D. Xu, "A reconfigurable smart sensor interface for industrial WSN in IoT environment," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1417–1425, May 2014.

[198] K. Ahmed and M. Gregory, "Integrating wireless sensor networks with cloud computing," in *Proc. 7th Int. Conf. Mobile Ad Hoc Sensor Netw.*, Beijing, China, Dec. 2011, pp. 364–366.

[199] K. Suto *et al.*, "A failure-tolerant and spectrum-efficient wireless data center network design for improving performance of big data mining," in *Proc. IEEE 81st VTC Spring*, Glasgow, U.K., May 2015, pp. 1–5.

[200] *CISCO 819 Services Router*. Accessed: Mar. 20, 2017. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/routers/access/800/819/software/configuration/Guide/819_SCG/1_overview.html
[201] *Data in Motion (DMo)*. Accessed: Mar. 20, 2017. [Online]. Available: https://developer.cisco.com/site/data-in-motion/index.gsp
[202] M. Aazam, P. P. Hung, and E. N. Huh, "M2M Emergency help alert mobile cloud architecture," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Gwangju, South Korea, Mar. 2015, pp. 500–505.
[203] J. Ribault and G. Wainer, "Simulation processes in the cloud for emergency planning," in *Proc. 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput. (CCGrid)*, Ottawa, ON, Canada, May 2012, pp. 886–891.
[204] Z. Alazawi, S. Altowaijri, R. Mehmood, and M. B. Abdljabar, "Intelligent disaster management system based on cloud-enabled vehicular networks," in *Proc. IEEE 11th Int. Conf. ITS Telecommun.*, St. Petersburg, Russia, Aug. 2011, pp. 361–368.
[205] J. S. Preden *et al.*, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, Jul. 2015.
[206] J. Preden, J. Llinas, G. Rogova, R. Pahtma, and L. Motus, "On-line data validation in distributed data fusion," in *Proc. SPIE Ground/Air Multisensor Interoperability Integr. Netw. Persistent*, May 2013, Art. no. 87420S.
[207] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Syst. J.*, vol. 9, no. 2, pp. 350–365, Jun. 2015.
[208] A. M. K. Cheng, "Cyber-physical medical and medication systems," in *Proc. IEEE 28th Int. Conf. Distrib. Comput. Syst. Workshops*, Beijing, China, Jun. 2008, pp. 529–532.
[209] I. Lee and O. Sokolsky, "Medical cyber physical systems," in *Proc. ACM 47th Design Autom. Conf.*, Anaheim, CA, USA, 2010, pp. 743–748.
[210] R. Colomo-Palacios, V. Stantchev, and A. Rodríguez-González, "Special issue on exploiting semantic technologies with particularization on linked data over grid and cloud architectures," *Elsevier Future Gener. Comput. Syst.*, vol. 32, pp. 260–262, Mar. 2014.
[211] *ETSI–Intelligent Transport*. Accessed: Mar. 20, 2017. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/intelligent-transport
[212] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in *Proc. 2nd Int. Conf. Ad Hoc Netw. (ADHOCNETS)*, Victoria, BC, Canada, Aug. 2010, pp. 1–16.
[213] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.
[214] J. Zhang *et al.*, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
[215] X. Hu *et al.*, "Towards a service centric contextualized vehicular cloud," in *Proc. 4th ACM Int. Symp. Develop. Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Montreal, QC, Canada, Sep. 2014, pp. 73–80.
[216] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric Internet of Things services," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Madrid, Spain, Jun. 2015, pp. 1–2.
[217] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular ad-hoc networks: Paradigms, scenarios, and issues," *J. China Universities Posts Telecommun.*, vol. 23, no. 2, pp. 56–96, Apr. 2016.
[218] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Trans. Cloud Comput.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/document/7448886/
[219] P. P. Jayaraman *et al.*, "Scalable energy-efficient distributed data analytics for crowdsensing applications in mobile environments," *IEEE Trans. Comput. Soc. Syst.*, vol. 2, no. 3, pp. 109–123, Sep. 2015.
[220] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the Internet of Vehicles," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 133–144, Apr. 2015.
[221] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proc. 1st Int. Workshop Mobile Cloud Comput. Netw. (MobileCloud)*, Bengaluru, India, 2013, pp. 19–26.
[222] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "A scalable SDN slicing scheme for multi-domain fog/cloud services," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Bologna, Italy, Jul. 2017, pp. 1–6.

Mithun Mukherjee (S'10–M'16) received the B.E. degree in electronics and communication engineering from the University Institute of Technology, Burdwan University, India, in 2007, the M.E. degree in information and communication engineering from the Indian Institute of Science and Technology, Shibpur, in 2009, and the Ph.D. degree in electrical engineering from the Indian Institute of Technology Patna in 2015. He is currently a specially assigned Researcher with the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology. His research interests include wireless communications, energy harvesting, and fog computing. He was a recipient of the 2016 EAI WICON, the 2017 IEEE SigTelCom Best Paper Award, and the 2018 IEEE Systems Journal Best Paper Award. He was a Guest Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE ACCESS, *ACM/Springer Mobile Networks & Applications*, and *Sensors*. He has been serving as the Special Issue Editor of the *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*.

Lei Shu (M'07–SM'15) is currently a Lincoln Professor with the University of Lincoln, U.K., and a Distinguished Professor with Nanjing Agricultural University, China. He is also the Director of NAU–Lincoln Joint Research Center of Intelligent Engineering, China. His main research field is wireless sensor networks. He has authored over 300 papers in related conferences, journals, and books in the area of sensor networks. He was a recipient of the Globecom 2010 and ICC 2013 Best Paper Awards. He has served as an Editor-in-Chief for *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, and as an Associate Editor for the IEEE SYSTEMS JOURNAL and IEEE ACCESS. He has served as the Co-Chair for over 50 international conferences and workshops, e.g., IWCMC, ICC, ISCC, ICNC, and Chinacom, including roles as a Symposium Co-Chair for IWCMC 2012 and ICC 2012, the General Co-Chair for Chinacom 2014, Qshine 2015, Collaboratecom 2017, and Mobiquitous 2018, a Steering and TPC Chair for InisCom 2015, and a TPC member of over 150 conferences, e.g., ICDCS, DCOSS, MASS, ICC, Globecom, ICCCN, WCNC, and ISCC.

Di Wang received the M.S. degree in software engineering from the College of Information and Computer Engineering, Northeast Forestry University, Harbin, China, in 2017. His current research interests include edge computing and mobile cloud computing.