1

Mobile Fog Computing

Chii Chang, Amnir Hadachi, Jakob Mass, and Satish Narayana Srirama

Institute of Computer Science, University of Tartu, Estonia

1.1 Introduction

The Internet of Things (IoT) paradigm motivates various next-generation applications in the domains of smart home, smart city, smart agriculture, smart manufacturing, smart mobility, and so forth [1], where the online systems are capable of managing physical objects, such as home appliances, public facilities, farming equipment or production line machines via the Internet. Moreover, mobile objects, such as land vehicles (e.g. cars, trucks, buses, etc.), maritime transports (e.g. ships, boats, vessels, etc.), unmanned aerial vehicles (UAVs; e.g. drones), and user equipment (UE; e.g. smartphones, tablets, mobile Internet terminals, etc.), have become the indispensable elements in IoT to assist a broad range of mobile IoT applications.

Mobile IoT applications emphasize the connectivity and the interoperability among the IoT infrastructure and the mobile objects. For example, in an Internet of Vehicles (IoVs) application [2], the IoT-based smart traffic infrastructure provides the connected roadside units (RSUs) that assist the smart cars to exchange the current traffic situation of the city center toward reducing the chance of traffic accidents and issues. As another example, classic disaster recovery activities of a city require numerous manned operations to monitor the disaster conditions, which involve high risk for human workers. Conversely, by integrating an Internet of Drones (IoD) [3], the smart city government can dispatch a number of drones to monitor and to execute the tasks without sending human workers to the frontline. Unexceptionally, mobile IoT also has benefited maritime activities in terms of improving the information exchange among the vessels and the central maritime management system, hastening the overall process speed of fishery or marine scientific activities [4].

Fog Computing: Theory and Practice, First Edition.
Edited by Assad Abbas, Samee U. Khan, and Albert Y. Zomaya.

Besides the public applications, mobile IoT plays an important role in personal applications, such as Internet of Medical Things (IoMT) applications [5], which utilize both inbuilt sensors of the UE (e.g. smartphone) and the UE-connected body sensors attached on the patient to collect health-related data and forward the data to the central system of the hospital via the mobile Internet connection of the UE.

Explicitly, the mobile IoT applications described above are time-critical applications that require rapid responses. However, the classic IoT system architecture, which relies on the distant central management system to perform the decision making, has faced its limitation to achieve the timely response due to latency issues deriving from the dynamic network condition between the front-end IoT devices and the back-end central server. Furthermore, the large number of connected mobile IoT devices have raised the challenges of mobile Big Data [6] that increase the burden of the central server and hence, lead to bottleneck issues. In order to improve the agility and to achieve the goal of ultra-low latency, researchers have introduced fog computing architecture [1].

Fog computing architecture (the fog) distributes the tasks from the distant central management system in the cloud to the intermediate nodes (e.g. routers, switches, hubs, etc.), which contain computational resources, to reduce the latency caused by transmitting messages between the front-end IoT devices and the back-end cloud. Specifically, the fog provides five basic mechanisms: storage, compute, acceleration, networking, and control toward enhancing IoT systems in five subjects: security, cognition, agility, low latency, and efficiency [1]. For example, in IoV application, the central server can migrate the best route determination function from the cloud to the roadside fog nodes to assist the travel of the connected vehicles. As another example, in an outdoor-based IoMT application, the hospital system can distribute the health measurement function and the alarm function to the UE in order to perform timely determination of the patient's health condition and to perform an alarm to catch the proximal passengers' attention when the patient is having an incident.

Here, we use the term *mobile fog computing* (MFC) to describe the fog-assisted mobile IoT applications.

MFC brings numerous advantages to mobile IoT in terms of rapidness, ultra-low latency, substitutability and sustainability, efficiency, and self-awareness. However, the dynamic nature of MFC environment raises many challenges in terms of resource and network heterogeneity, the mobility of the participative entities, the cost of operation, and so forth. In general, the static fog computing frameworks designed for applications, such as the smart home or smart factory would not fully address the MFC-specific challenges because they have different perspectives from the involved entities and the topology. For example, a classic fog computing framework, which may involve a thin mobile client-side application for smartphone users, would not consider how to provide a reliable fog service to the high-speed moving vehicles. Moreover, the classic fog computing framework also would not consider how to provide a reliable fog service to vessels at sea where the telecommunication base stations are not available, and the satellite Internet is too expensive.

The goal of this chapter is to provide an introduction and guidance to MFC developments. Specifically, different from the existing works [7, 8] related to MFC, this chapter discusses MFC in four major application domains: land vehicular fog computing (LV-Fog), marine fog computing (Marine Fog), unmanned aerial vehicular fog computing (UAV-Fog), and user equipment-based fog computing (UE-fog).

The rest of the chapter is organized as follows: Section 1.2 clarifies the term *MFC*. Section 1.3 breaks MFC down into four application domains and describes their characteristics. Section 1.4 enlists the wireless communication technologies used in the mentioned application domains, while Section 1.5 proposes a taxonomy of nonfunctional requirements for MFC. Open research challenges, both domain-oriented and generic, are identified in Section 1.6, and finally, Section 1.7 concludes this chapter.

1.2 Mobile Fog Computing and Related Models

In this chapter, MFC has its specific definition and it is not an exchangeable terminology with the other similar terms, such as mobile cloud computing (MCC) or multi-access (mobile) edge computing (MEC). In order to clarify the meaning of MFC, one needs to understand the aspects of the parties who introduced or adapted the terminologies. Commonly, MCC refers to a system that assists mobile devices (e.g. smartphones) to offload their resource-intensive computational tasks to either distant cloud [7] or to the proximity-based cloudlet [8].

Fundamentally, MEC is an European Telecommunications Standards Institute (ETSI) standard aimed to introduce an open standard for telecommunication service providers to integrate and to provide infrastructure as a service (IaaS), platform as a service (PaaS), or software as a service (SaaS) cloud services from the industrial integrated routers or switches of their cellular base stations. Explicitly, MEC is an implementation approach rather than a software architecture model. Further, as stated by ETSI, ETSI and OpenFog are collaborating to enable the MEC standard and the OpenFog Reference Architecture standard (IEEE 1934) to complement each other [9].

Today, researchers of industry and academia have been broadly using *edge computing* as the exchangeable term with fog computing. However, National Institute of Standards and Technology (NIST) and the document of IEEE 1934 standard for fog computing reference architecture, which derives from OpenFog Consortium, have specifically explained the differences between fog computing and edge computing. Accordingly, “the Edge computing is the network layer encompassing the end-devices and their users, to provide, for example, local computing capability on a sensor, metering or some other devices that are network-accessible” [10]. Further, based on the literature in edge computing domain, which include cloudlet-based computing models [11], one can explicitly identify that edge computing is loosely a *bottom-up* model. Specifically, an edge computing-integrated system emphasizes task offloading from the end-devices to the nearby cloudlet resources, which are capable of providing Virtual Machine (VM) or containers engine (e.g. Docker[1])-based service to the other nodes within the same subnet.

On the other hand, fog computing is a hierarchical *top-down* model in which the system specifically tackles the problem about how to utilize the intermediate networking nodes between the central cloud and the end-devices to improve the overall performance and efficiency. Commonly, such intermediate nodes are Internet gateways such as routers, switches, hubs (e.g. an adaptor that interconnects Bluetooth-based device to IP network). Moreover, a fog node is capable of providing five basic services – storage, compute, acceleration, networking, and control [1]. Correspondingly, when a cloudlet or an IoT device is providing gateway mechanism to the other nodes and they are capable of providing some or all of the basic fog services, we also consider them as fog nodes.

By extending the notion above, MFC is the subset of fog computing that addresses mobility-awareness. Specifically, MFC involves two types – infrastructural fog (iFog)-assisted mobile application and mobile fog node (mFog)-assisted application. In summary, iFog-assisted mobile application enhances the performance of a cloud-centric mobile application by migrating the processes from the central cloud to the stationary fog nodes (e.g. the cellular base station) that are currently connecting with the mobile IoT devices. On the other hand, mFog-assisted applications host fog services on mobile gateways (e.g. in-vehicle gateway devices or smartphones) that interconnect other devices (e.g. onboard sensors, body sensors, etc.) or other things (e.g. proximal cars, people, sensors, etc.) to the cloud.

1.3 The Needs of Mobile Fog Computing

MFC encompasses four application domains: land vehicular applications, marine applications, unmanned aerial vehicular applications, and UE-based applications. Specifically, each domain involves both iFog- and mFog-based architecture. Ideally, the approaches of iFog aim to provide generic solutions that are applicable

[1] www.docker.com.

to all the MFC domains where the infrastructure is applicable. On the other hand, mFog-based approaches aim to overcome the challenges in which the iFog is inapplicable or is unable to resolve effectively. To clarify the terminologies used in the rest of the chapter, iFog denotes infrastructural fog node and mFog represents the generic term of mobile fog nodes. Moreover, we further classify mFog to four types: LV-Fog, Marine Fog, UAV-Fog, and user equipment-based fog (UE-fog) corresponding to the mobile fog node hosted on a land vehicle, a vessel, a UAV, and a UE (e.g. smartphone, tablet, etc.).

1.3.1 Infrastructural Mobile Fog Computing

1.3.1.1 Road Crash Avoidance

The number of vehicles on the road is increasing every year as well as the number of road accidents [12]. In order to reduce or avoid the collision accident, academic and industrial researchers have been working on improving the safety aspect of the vehicles. Specifically, the advancement in communication technologies has allowed the development of advanced driver-assistance systems (ADAS), which has emerged as an active manner of preventing car crashes. ADAS has made many achievements through the development of systems that include rear-end collision avoidance and forward collision warning (FCW). The vehicle-to-vehicle (V2V) communication plays a big role in ADAS systems and it is manifested in ensuring that the controllers on board the vehicles (i.e. onboard unit, OBU) are capable of communicating with other vehicles for the purpose of negotiating maneuvers in the intersections and applying automatic control when it is necessary to avoid collisions [13]. The success of these systems relies a lot on the reliability of the communication. Therefore, many models of V2V communication have been investigated. Some of them focus on probabilities and analytic approaches in modeling the communication message reception while others adapt Markovian methods to assess the performance and reliability of the safety-critical data broadcasting in IEEE 802.11p vehicular network [14]. Vehicular ad-hoc network (VANET) has also contributed to integrate and improve the car-following model or platooning, which reduces the risks of collisions and makes the driving experience safer [15]. Explicitly, today's smart vehicular network systems have applied the fog computing mechanisms that utilize the cloud-connected OBUs of the vehicle to process the data from the onboard sensors toward exchanging context information in the vehicular network and participating in the intelligent transport systems.

1.3.1.2 Marine Data Acquisition

Today, marine data acquisition and cartography systems can achieve low-cost data acquisition and processing by composing IoT, mobile ad hoc network (MANET), and delay-tolerant networking (DTN) technologies. Specifically, sea vessels, which equip multiple sensors, can utilize International Telecommunication Union (ITU) standards-based very high frequency (VHF) data exchange system to route the sensory data to the gateway node (i.e. cellular base station) at the shore via the ship-based MANET. Afterwards, the gateway can relay the data to the central cloud. In general, such an architecture may produce many duplicated sensory transmission readings due to the redundant data transmitted from different ships. In order to remove such duplication and to improve the efficiency, the system can deploy fog computing service at the gateway nodes to preprocess the sensory data toward preventing the gateways sending duplicated data to the central server [4].

1.3.1.3 Forest Fire Detection

Emerged smart UAVs, which are relatively inexpensive and can be flexibly dispatched to a large area under different weather conditions, both during day and night, without human involvement are the ideal devices to handle forest fire detection and firefighting missions. Specifically, with onboard image detection mechanism and mobile Internet connectivity, UAVs can provide real-time event reporting to the distant central management system. Further, in order to extend the sustainability of the image-based sensing mission, the system can distribute the computational image detection program to the proximal iFog hosted on cellular base stations and made accessible via standard communication technologies, such as Long-Term Evolution (LTE), SigFox, NB-IoT, etc. Hence, the UAVs can use their battery power only for flying and sensing tasks [16] (Figure 1.1).

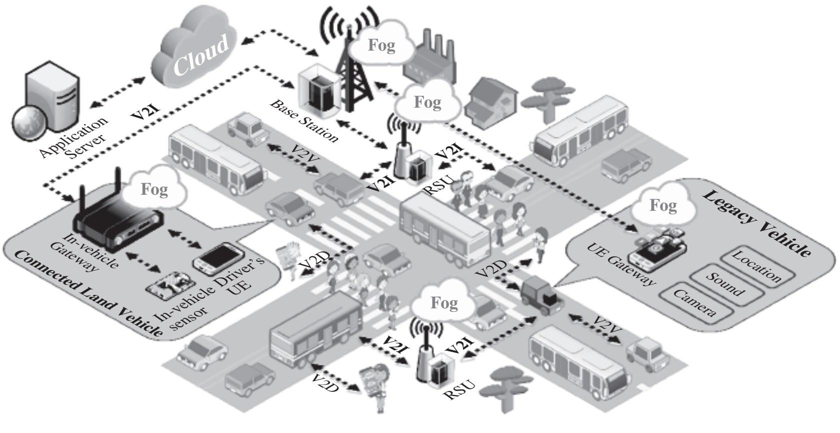

Figure 1.1 Land-vehicular fog computing examples. (*See color plate section for the color representation of this figure*)

1.3.1.4 Mobile Ambient Assisted Living

Today's UE devices, such as smartphones, have numerous inbuilt sensor components. For example, the modern mobile operating systems (e.g. Android OS) have provided numerous software components that are capable of integrating both internal and external sensors to support mobile Ambient Assisted Living (AAL) applications such as real-time health monitoring and observing the surrounding environments of the user to avoid dangers. Fundamentally, classic mobile AAL applications rely on the distant cloud to process the sensory data in order to identify situations. However, such an approach is often unable to provide rapid response due to communication latency issues. Therefore, utilizing proximal fog service derived either from the MEC-supported cellular base station or individual or small business-provided Indie Fog [17] has become an ideal solution to enhance the agility of mobile AAL applications [18].

1.3.2 Land Vehicular Fog

The development of vehicular networking has improved safety and control on the roads. Especially, LV-Fog nodes have emerged as a solution to introduce computational power and reliable connectivity to transportation systems at the level of Vehicle-to-Infrastructure (V2I), V2V, and Vehicle-to-Device (V2D) communications [19]. These networks are shaped around moving vehicles, pedestrians equipped with mobile devices, and road network infrastructure units. Further, these aspects have facilitated the introduction of real-time situational/context awareness by allowing the vehicle to collect or process data about their surroundings and share these insights with the central traffic control management units or other vehicles and devices in a cooperative manner.

To perform such activities, there is a need for adequate computing resources at the edge for performing time-critical and data-intensive jobs [20] and face all the challenges related to data collection and dissemination, data storage, mobility-influenced changing network structure, resource management, energy, and data analysis [21, 22].

Most of the techniques proposed to solve these challenges are focusing on merging the computation power between vehicular cloud and vehicular networks [19]. This combination allows usage of both vehicles' OBUs and RSUs as communication entities. Another side that has been investigated is the issues related to latency and quality optimization of the tasks in the Vehicular Fog Computing (VFC) [20], and it was formulated by presenting the task as a bi-objective minimization problem, where the trade-off is preserved between the latency and quality loss. Furthermore, handling the mobility complexity that massively affects the network structure is addressed by using mobility patterns of the moving vehicles and devices to perform a periodic load balancing in the fog servers [23] or distance-based forwarding (DBF) protocol [19]. The energy management and computational power for data analysis are controlled by distributing the load among the network entities to make use of all the available resources based on CR-based access protocol [22].

Moreover, the design of the Media Access Control (MAC) layer protocol in the vehicular networks is essential for improving the network performance, especially in V2V communication. V2V enables cooperative tasks among the vehicles and introduces cooperative communication, such as:

Dynamic fog service for next generation mobile applications. The emergence of new mobile applications, such as augmented reality (AR) and virtual reality, have brought a new level of experience that is greedy for more computational power. However, the traditional approach of a distant cloud-driver is incapable of achieving with good performance due to latency. Therefore, introducing *Metropolitan vehicle-based cloudlet*, which is a form of mobile fog node model, solves the latency issue by dynamically placing the fog at the areas with high demand. Furthermore, by adopting a collaborative task offloading mechanism, the vehicle-based mobile fog nodes are capable of effectively distributing the processes across all the participative nodes, based on their encounter conditions [24].

Federated intelligent transportation. Traffic jams start to have a considerable negative impact by wasting time, fuel, capital, and polluting the environment due to the nonstop increase in the number of vehicles on the roads [25]. Fortunately, cloud-driven smart vehicles have emerged as a facilitator to overcome the problem. The solution resides in considering the serviceability level of mobile vehicular cloudlets (MVCs), which are a form of the mobile fog node model, based on the real-world large-scale traces of mobility of urban vehicles collected by onboard computers. Based on the peer-to-peer communication network, vehicles can further improve the traffic experience by exchanging real-time information and providing assistance to the manned or unmanned vehicles [26].

Vehicular opportunistic computation offloading. Public transportation service vehicles, such as buses and trams, which commonly have fixed routes and time schedules, can be the mobile fog nodes for the other mobile application devices inside the proximal encountered vehicles that need to execute time-sensitive and computation-intensive tasks, such as augmented reality (AR) processes used for the advanced driver assistance systems and applications [27].

1.3.3 Marine Fog

Integrating IoT to existing legacy marine systems can provide rapid information exchange. Initially, classic marine communication systems utilize VHF radio to obtain ship identification, ship location, position, destination, moving speed, and so forth. Unfortunately, VHF can provide only 9.6 kbps, which is insufficient to provide marine sensory data streaming [28]. Alternatively, ships can utilize the new satellite Internet to deliver their data, which is capable of achieving 432 kbps. However, satellite communication is not affordable for small and medium-size businesses since a simple voice service can cost USD $13.75 per minute [28]. In order to overcome the issue, researchers have introduced fog computing and networking-integrated marine communication systems for the Internet of marine t hings (IoMaT) [4]. Here, we term such a fog computing model Marine Fog.

By integrating a virtualization or containerization technology-based fog server with onboard equipment, vessels are capable of realizing a software-defined network (SDN) that allows the vessels to (re)configure a message routing path dynamically. Afterwards, by utilizing Marine Fog–based SDN mechanism, vessels can easily establish an ad hoc–based DTN, which caches data at the onboard Marine Fog node until the vessel encounters the next Marine Fog node, for delivering sensory data from the data source to the base stations toward relaying the data to the cloud. Moreover, an advanced Marine Fog node within the network may also perform data preprocessing in order to further reduce the transmission latency [29] (Figure 1.2).

Existing wireless sensor network (WSN) architecture in marine monitoring uses sea buoys as sink nodes, capable of communicating with nearby sensor nodes (other buoys, vessels) directly (e.g. using ZigBee), as well as via the cellular Internet network [30]. By introducing the previously mentioned virtualization, the WSN architecture could be extended to be used in Marine Fog. However, this approach amplifies the need for energy-harvesting technology at the buoys.

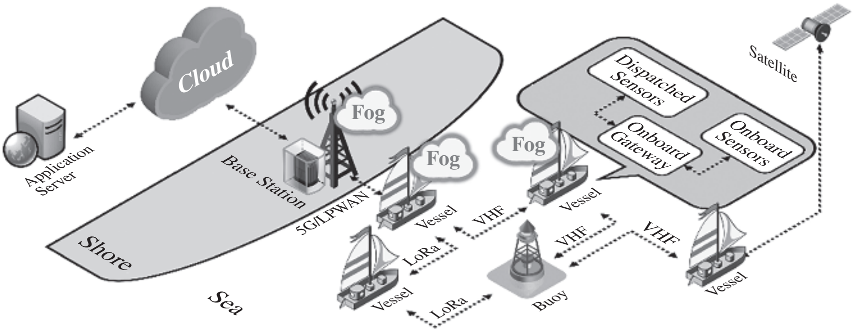

Figure 1.2 Maritime fog computing examples.

1.3.4 Unmanned Aerial Vehicular Fog

UAVs, which are also referred to as drones, can be employed in a broad range of applications due to their unrestricted geo-location feature. Hence, they have become one of the major elements in the IoT ecosystem. For example, smart city application can utilize UAVs as mobile sensors that can be sent to critical environments that are dangerous for humans (e.g. a forest fire) [16]. Specifically, by utilizing UAVs, the system is capable of establishing a UAV ad hoc network that is capable of performing a wide-ranged geolocation-based sensory data streaming network dynamically. Moreover, by mounting fog nodes on UAVs to perform sensory data analysis, the system can further improve the agility of identifying the emergency situations.

Below, we describe the features of the UAV-Fog [31], with corresponding examples indicated in Figure 1.3.

Fast deployment. Modern UAVs are capable of carrying on tasks programmatically without human interference. Further, a system can dispatch a large number of UAVs to perform a temporary mission in an area where the manned vehicles are unable to reach or unable to effectively perform the tasks. For example, UAV-Fogs can assist wildfire problems in Portugal, Spain, and Australia [32].

Scalability. The rapid growth of large-scale IoT applications requires more network infrastructure and fog computing resources in order to compensate for ultra-low latency. However, investing in the base infrastructure in certain areas is not cost-efficient. Hence, instead of developing the infrastructural IoT and fog network, the service provider can deploy UAVFog nodes to those areas. For example, in order to support the sensory data streaming performed by underwater vehicles, UAV-Fog nodes can fly above the water surface in order to route the data stream to the base station at the shore [31].

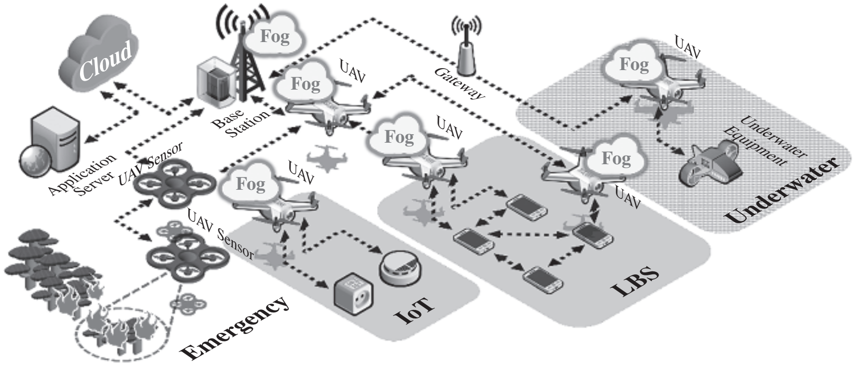

Figure 1.3 UAV fog computing examples.

Flexibility. UAV-Fog nodes can equip heterogeneous capabilities to support various applications. For example, an Olympic event in a city lasts 16 days. During the contests, a large number of visitors are gathering in the city and many of them are using Social Network Services (SNS) to disseminate information (e.g. text, image, video posts) related to the event. However, the city's network infrastructure may not have sufficient capacity to provide the high-quality experience for the SNS users due to traffic overload. In order to support the best quality of experience (QoE) for the SNS users, SNS providers may deploy UAV-Fog nodes to the city to provide a temporary location-based social network (LBSN) mechanism that directly routes the content (e.g. Twitter posts, YouTube video stream, etc.) within the city when the content provider and the receiver are within the city.

Cost-effective. The content described in previous paragraphs has indicated that employing UAV-Fog nodes is a cost-effective solution for many applications that require only a temporary enhancement for computational or networking needs. For example, wildfires in Australia often occur in areas where the network infrastructure is unavailable. Hence, establishing an infrastructural IoT-based smart monitoring system at such an area is unrealistic. Second, many cities in the world are unable to provide fundamental infrastructure for the rapid growth of IoT applications. Instead of waiting for the hardware service provider to complete the infrastructure, the IoT software service provider can simply deploy more UAV-Fog nodes to the areas that require more resources. Finally, many cities often spent a large amount of money on network infrastructure for temporary events, which is cost-inefficient. Although it is possible to send the manned land vehicular-based nodes (e.g. mobile base stations) to support the need, compared to unmanned UAVs, the manned solutions require payroll for human workers and extra petrol or electricity, since the movement of land vehicles is constrained based on the roads.

1.3.5 User Equipment-Based Fog

UE represents the end-users' terminal devices (e.g. smartphones) that are connecting to an Internet Service Provider (ISP)'s network. Initially, UEs are thin clients in the IoT ecosystem. However, recent research efforts have utilized UEs as one of the major service provisioning elements. Accordingly, the following use cases illustrate the usefulness of fog-integrated UEs (Figure 1.4).

1.3.5.1 Healthcare

Healthcare is one of the top-five application domains in fog computing that has potential market value up to $2737 billion by year 2022 [33]. Specifically, IoMT applications, which commonly rely on the central cloud for managing data and

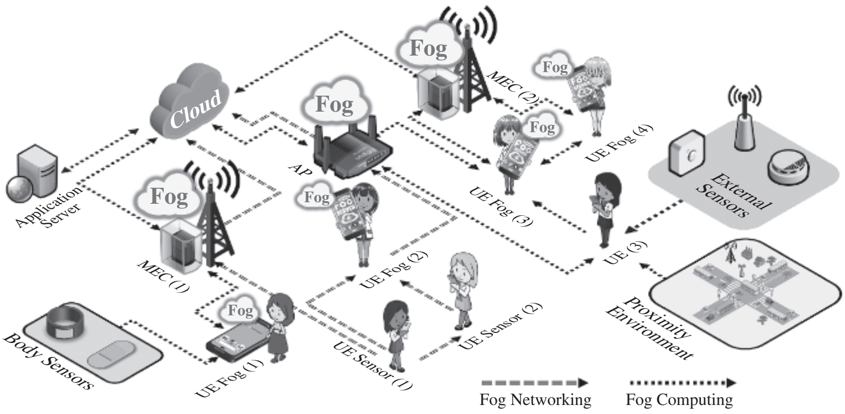

Figure 1.4 UE fog computing examples. (*See color plate section for the color representation of this figure*)

performing decision-making, are now distributing certain tasks to the intermediate gateways. In particular, IoMT applications broadly utilize UEs (e.g. smartphones, tablets, etc.) as the gateways of wearable body sensors, in order to let the IoMT servers (e.g. hospital) acquire the sensory data anytime, anywhere from the patients. Further, considering the need of agile sensory data stream processing when the patients are in outdoor areas, which may not be able to maintain a high-quality network connection, utilizing the infrastructural fog (e.g. hosted at the cellular base station or ISP's Wi-Fi access points [APs]) can highly improve the overall agility of the data processing and identification of emergency situations. Moreover, considering modern UEs have powerful central processing units (CPUs) and decent storage spaces when the infrastructural fog is unavailable, the processes can also migrate to UEs to support the service continuity [34].

1.3.5.2 Content Delivery

Besides utilizing UEs as computing-based fog servers, UE-based fog computing nodes (UE-fog nodes) are also the ideal networking service providers. For example, replays of soccer matches, which catch important moments of the game, haver become an indispensable element of the sports game for the fans both inside or outside the stadium. Fundamentally, the audience members download the replay video to their UEs via the Internet. However, considering that there can be a large number of requests coming from the audience, the local wi-fi or cellular base stations may not be able to provide sufficient speed, especially when the audience demands high-quality or even ultra-high-quality videos. In order to address this problem, the application can host fog servers in the UEs of the audience to support the SDN mechanism, which allows the service provider to establish a local ad-hoc content caching group among the fans in the stadium and its surrounding areas toward reducing the burden of both the ISP and the soccer replays service provider [35].

1.3.5.3 Crowd Sensing

Integrating UEs into the fog can raise many advanced people-centric applications that are directly related to people's daily lives. For example, in the lost child situation, local government or police can collaborate with a local ISP to request the SDN mechanism-supported UE-fog nodes, which are hosted on the smartphones of the people on-site, to assist the lost child situation by utilizing the inbuilt cameras and audio recorder of the UEs as the sensors, then utilize some of the more powerful UE-fog nodes as the super-peers to route the data to the local cloud of the ISP, which is accessible by the government and the police, toward hastening the entire process. Moreover, the UE-fog nodes that have high-performance computational power can also provide context as a service (CaaS) instead of routing the raw sensory data [36]. Specifically, CaaS mechanism–supported UE-fog nodes allow the requesters to deploy their own data processing algorithm to UE-fog nodes toward preprocessing the raw sensory data and hastening the overall process.

1.4 Communication Technologies

In this section, we give an overview of communication technologies used by mobile things and mobile fog nodes in each of the major application domains. Unsurprisingly, in existing literature Wi-Fi technology is most prevalent, due to the ready availability of devices with 802.11 support: routers, smartphones, single board computers. Wi-fi is suitable for mFog thanks to the easy mobility of Wi-Fi AP technology. On the other hand, 4G technology, e.g. long-term evolution (LTE), which needs static base stations is the dominant option for iFog.

The fast movement of nodes among road network environment has obliged the existing protocol for the physical layer to consider multipath fading and Doppler frequency shifts. Therefore, the trend is to use very high-frequency radio waves, such as micro or millimeter waves.

1.4.1 IEEE 802.11

The IEEE 802.11 set of specifications, commonly referred to as Wi-Fi, is the most widely used wireless communication technology found in fog computing. Since Wi-Fi infrastructure is widely deployed in homes, offices, public spaces, and so forth, it is the natural choice for fog-thing and fog-UE communication, with the fog node hosting the Wi-Fi AP.

The standards *802.11n* and *802.11ac*, for instance, have a typical data rate of 200 and 400–700 Mbps respectively, the typical range of 802.11 routers in the 2.4 GHz band can be up to 50 m [37]. The next generation *802.11ax* promises to enhance data rates further, but it is unlikely that significantly higher coverage range will be achieved in practice, due to the recent versions using the 2.4 and 5 GHz bands.

Interestingly, advertising capabilities of 802.11 can be improved by including additional information (e.g. fog node capability and status) in the 802.11 advertising beacons [38].

The mentioned signal coverage ranges are suitable in domains where the client device mobility speed is low; consider, for example, pedestrians in UE-fog. Additionally, the smartphones already employ the technology. Existing UE-fog research that does not include real-world technology choice and simply consider the data rate aligns with the capabilities of wi-fi. For example, even 6.9 Gbps rates [39] are theoretically supported by *802.11ac*. In terms of existing research prototypes, laptop hotspots are a common choice to establish the wi-fi AP [40–42]. Since laptop hotspots generally operate with *802.11n* technology, it is important to consider the newer standards in future fog prototypes.

In UAV-fog, following the mFog concept, a wi-fi AP could reside at UAV node or, alternatively, static *802.11ac* APs may act as sink nodes supporting a group of UAVs [16].

IEEE 802.11p, a.k.a. wireless access in vehicular environments (WAVEs) is adapted for the wireless environment with vehicles. In addition, they are designed in such manner that they are very suitable for single hop broadcast V2V communications; however, this technology suffers from an issue related to scalability, reliability, and unbounded delays due to its contention-based distributed medium access control mechanism [23, 43].

For maritime use cases, the coverage ranges offered by Wi-Fi are generally not suitable. However, Wi-Fi is useful in vessels for onboard networks where the clients are crew and passengers, but such scenarios can be categorized rather as UE-fog.

1.4.2 4G, 5G Standards

Currently operating cellular networks target the requirements of the 4G standard (also known as IMT-Advanced), specified by the International Telecommunication Union (ITU) in 2008 [44]. For instance, the requirements suggest 100 Mbps data rates for clients moving at high speeds (e.g. in a train) and 1 Gbps for stationary situations.

Among technology standards accepted as fulfilling the 4G requirements are IEEE *802.16m* (WiMAX v2) and 3 GPP Long Term Evolution-Advanced

(LTE Advanced), the latter of which has seen far bigger deployment and thus is the common option for existing mobile fog.

To supersede 4G, the ITU is defining requirements for the 5G networks, also called IMT-2020. The 2017 draft of technical performance requirements [45] notes peak data rates of 20 and 10 Gbps for downlink and uplink, respectively, and specifies channel link data rates for four different mobility classes. 5G also specifically targets supporting cases where the density of devices is large, growing from 4G's 10^5–10^6 devices km^{-2} [46].

5G networks are enablers for smart collaborative vehicular network architecture since they provide possibilities for fulfilling the requirements of reliability, handover, and throughput of future vehicular networks [47]. LTE D2D-based VANET has proven to be suitable for the safety-critical IoV applications, thanks to their effectiveness in coping with high mobility and precise geo-messaging [43].

The MEC paradigm has introduced the handover and migration of VMs to the cellular base stations for supporting the UE [48–50], however, the similar idea potentially applies to the other mobile fog domains.

In maritime fog systems, the shore-located cellular base stations can be leveraged to also act as sink nodes [4, 51], gathering sensor data from the vessels. Multiple access techniques, such as nonorthogonal multiple access (NOMA) offered by 5G, are considered useful for UAV cloudlets to maximize efficiency [52]. However, generally 4G/5G coverage is available in more urban areas, so for marine communication at sea or UAV deployments in remote areas, alternatives such as satellite communication need to be considered.

1.4.3 WPAN, Short-Range Technologies

From the perspective of the mobile thing, wireless personal area network (WPAN) technologies such as ZigBee (802.15.4) and Bluetooth (801.15.1) are suitable for lower bandwidth and lower energy communication needs, such as interacting with IoT devices or exchanging metadata.

The shorter range, while unfit for marine scenarios, can be applied in UAV-Fog since use cases,such as supporting land/marine vehicle, mandate that the UAV itself will adjust its location to stay close to the peers [31].

The traditional Wi-Fi AP-based infrastructure can be expanded using Wi-Fi direct. Here the client devices form a local Wi-Fi direct group, reducing the load on the AP by locally disseminating the data [35, 53] and advertising device services [54].

Bluetooth and Bluetooth Low Energy are common choices for UE when mediating data from other devices to a fog node (e.g. Wi-Fi AP-based), for instance, forwarding sensor data from Medical IoT devices [5].

Near-field communication (NFC) radio-frequency identification (RFID)-based technology, such as NFC may add an additional layer of physical security due to the extremely low signal range, for instance in UE-based Mobile-to-Mobile computation off-loading [55].

1.4.4 LPWAN, Other Medium- and Long-Range Technologies

VHF marine radio VHF is the internationally used technology for marine radio in the frequency range 156—163 MHz. Typical range of VHF is reported to be up to 70 nautical miles from a land-based station [56], while ship-to-ship signal range is below 40 km [57]. However, VHF is limited in supported data rate, which is below 30 kbps. While VHF may be sufficient to transmit sensory readings from ships to shore-deployed fog nodes [4] for aggregation and forwarding, the low data rate is unsuitable for agile transfer of larger data (e.g. video streams or VM images).

Satellite systems offer higher speeds compared to VHF and provide the greatest signal coverage, which is an important factor considering the distances involved in the marine domain. Yet, due to their high cost, these are a viable option only for larger vessels [57, 58].

Low-power wide-area networks (LPWANs) long range (LoRa) has been receiving attention lately as an energy-efficient, long-range wireless technology. In the mF2C project [29], the physical layer-LoRa, accompanied with LoRaWAN at the data link layer is used for ship-to-ship communication, while [59] use it for ship-to-shore communication in harbors. LoRa with LoRaWAN can cover up to 15 km in rural areas with a data rate up to 37.5 Kbps [60], making it a lower-energy alternative to VHF.

Sigfox and NB-IoT can be considered as competitors to LoRaWAN. While LoRaWAN and Sigfox operate in unlicensed bands, NB-IoT operates in licensed frequency bands.

Dedicated short-range communications (DSRC) is another technology defined especially for VANET, which are one-way or two-way short-range to medium-range wireless communications designed for allowing V2V and V2I communications. It is characterized by its frequency of 75 MHz licensed spectrum in 5.9 GHz band, which is provided by Federal Communications Commission (FCC) in the United States [20, 26].

1.5 Nonfunctional Requirements

An MFC system needs to address a number of nonfunctional requirements in order to achieve the basic quality of service (QoS) principles. In order to provide a comprehensive guide to the developers, this section describes the nonfunctional requirements in five aspects – heterogeneity, context-awareness, tenant, provider, and security.

Figure 1.5 summarizes the elements of the five aspects of the nonfunctional requirements. In general, the five aspects are corelated to one another. For example, heterogeneity is a common factor that needs to be considered when the fog tenant plans to choose which fog server to use to deploy their applications. Similarly, context awareness, which represents the runtime factors, is also influencing the decision of when and where to deploy and execute the tasks. Furthermore, the tenant-side end-device or end-user (i.e. tenant-side clients) is influencing the decision of how the provider of fog servers manage the fog nodes. On the other hand, the QoS of the provider's fog nodes also influences the decision for the tenant to choose the right fog node for tenant-side clients. In addition, although all the four MFC domains involve the five aspects, the complexity level of each aspect in a different domain can be quite different. For example, an application scheduling scheme designed for LV-fog may require significant adjustment when the developers intend to apply the scheme to UAV-fog because the heterogeneity level and the context factors are very different between the two domains.

As indicated above, we distinguish the tenant and the provider in which the *providers* represent the owners who own the fog server machines and are providing the fog infrastructure and PaaS to application service providers known as the *tenants* in a multitenancy manner [61]. For example, a telecommunication company such as Vodafone may host fog servers on their base transceiver station (BTS) and enable the accessibility of the fog server via 4G/5G/5G NR (https://www.qualcomm.com/invention/5g/5g-nr) communication. Therefore, application service providers can tenant the fog servers and deploy fog-based applications on the

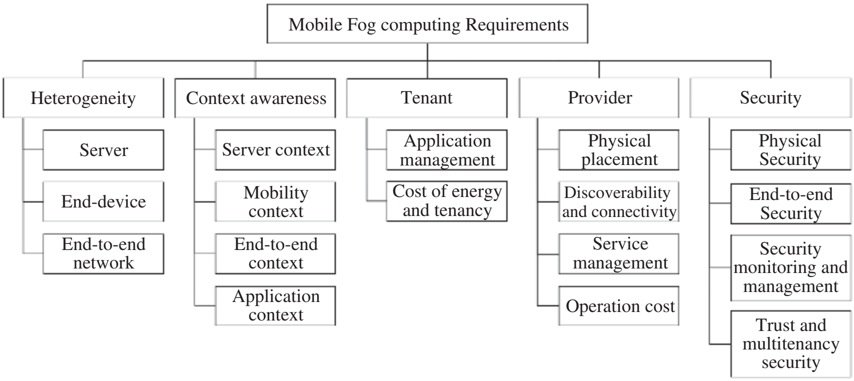

Figure 1.5 A taxonomy of non-functional requirements of mobile fog computing.

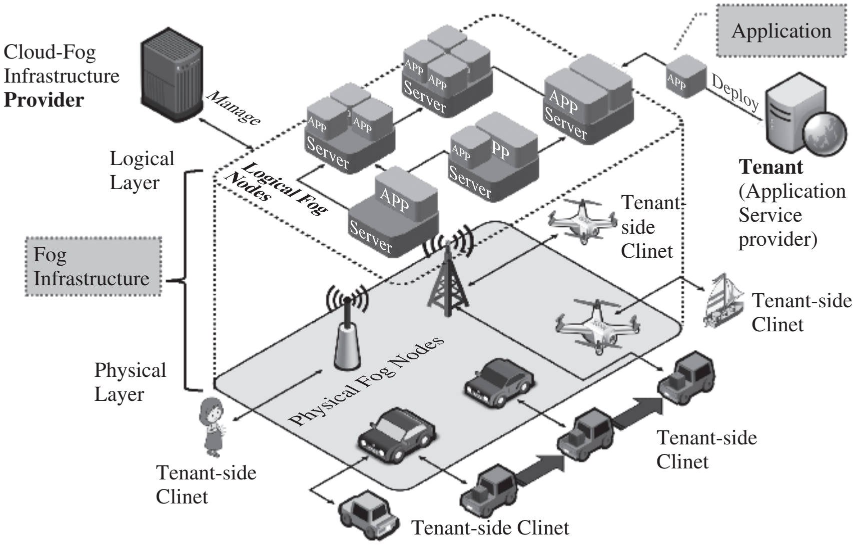

Figure 1.6 Fog infrastructure service provider, fog service tenant, and tenant-side clients.

servers toward serving the tenant-side clients. Correspondingly, Figure 1.6 illustrates the relationships among the involved entities.

Note that although, in a general perspective, fog servers are expected to support multitenancy [61], in many cases, service providers may provide the fog nodes in a highly integrated manner in which a single provider manages both the underlying fog servers and the application services. For example, it would be a common case that an indie fog [17]-based UE-fog service provider who follows the common standards to provide the micro-fog services from their own customer premises equipment (CPE) would manage both the fog service software and the host hardware. As another example, Marine Fog systems would deploy the integrated fog nodes on vessels based on the isolated platform for marine activities in order to prevent security issues.

Based on the state-of-the-art literature in both iFog and mFog across the four MFC domains, we explain the elements of the five aspects and what needs to be addressed in order to achieve the QoS in MFC.

1.5.1 Heterogeneity

There are three types of *heterogeneity*: server heterogeneity, end-device heterogeneity, and end-to-end heterogeneity.

1.5.1.1 Server Heterogeneity

Different to the common IaaS/PaaS-based cloud services, which are virtual resources, fog services are hosted on resource constrained physical equipment that has limited computational power and networking performance. Therefore, when tenants intend to deploy their applications, they need to consider the compatibility, connectivity, interoperability, and reliability of the fog servers. Specifically, we can classify the heterogeneity of the fog servers into two aspects: hardware type and software type.

- *Hardware type*. Represents the hardware component specification and configuration. In detail, the provider should clearly provide information on the hardware in terms of the computational resource specifications, such as CPU model code and speed, RAM model code and speed, read/write speed of storage, independent or integrated GPU, vision processing unit (VPU), field-programmable gate array (FPGA), application-specific integrated circuit (ASIC), AI accelerator, etc.; the available networking resources specification, such as IEEE 802.11a/b/g/n/ac, Bluetooth LE, IEEE802.15.4, LoRa, NB-IoT, etc.; extra components such as inbuilt or connected sensors that are accessible via the API provided by the fog server. Furthermore, if the fog server is hosting on a mobile Fog node, the provider should also provide the corresponding mobility-related information, such as the route of its moving path, the moving speed, and so forth.
- *Software type*. Denotes the software application deployment platform supported by the fog server. For example, the fog server may support VM-based service which allows a flexible configuration. Alternatively, the fog server may provide a containerization-based platform service, such as Docker (https://www.docker.com). Whereas, for the resource-constraint devices, which can serve only a limited number of requests, the provider may configure a FaaS-based platform that allows the tenant to deploy functions on the fog servers to provide microservice to tenant-side clients instead of the completed applications.

1.5.1.2 End-Device Heterogeneity

The heterogeneity in hardware specification and the software specification of the end-device/user influences the overall QoS and QoE that tenant can provide. In order to achieve the best QoS and QoE, tenants need to choose fog nodes that are most compliant and most efficient for their end-devices. Specifically, end-device heterogeneity involves hardware specification in terms of the type of the device (e.g. smartphone, mobile sensor, drone, etc.), and computational power and network capabilities, which influence how the end-device/user interact with the tenant's fog applications deployed on the provider's fog server. Further, from the software specification aspect, the tenants need to consider what software they can install at the end-device/user-side and how well the client-side software may perform when it interacts with the fog nodes?

1.5.1.3 End-to-End Network Heterogeneity

The end-to-end network heterogeneity involves the three aspects listed here, and all of them would influence the performance of the other nonfunctional requirements.

- *Device-to-fog (D2F)*. Represents the radio network and the communication protocol used between the fog server and the end-device. In general, D2F has a broad range of networking options and hence, increase the complexity from the perspective of both fog server provider and the tenant. For instance, radio network options encompass IEEE 802.11 series, IEEE 802.15.4, Bluetooth, 3G/4G cellular Internet, LoRa, SigFox, NB-IoT, LTE-M, 5G New Radio (5G NR), and so forth, which indicates that the provider may need to support multiple radio networks in order to fulfill multitenancy. On the other hand, tenants also need to consider the available radio network provided by the fog server in order to optimize their applications.
- *Fog-to-fog (F2F)*. Represents the communication network between fog nodes in both horizontal and vertical manner. For example, Figure 1.7 illustrates the vertical communication between two LV-Fog nodes, while both LV-Fog nodes also rely on an iFog node to interconnect them to the cloud. Explicitly, the F2F network would highly influence the performance of the tenant-side application,

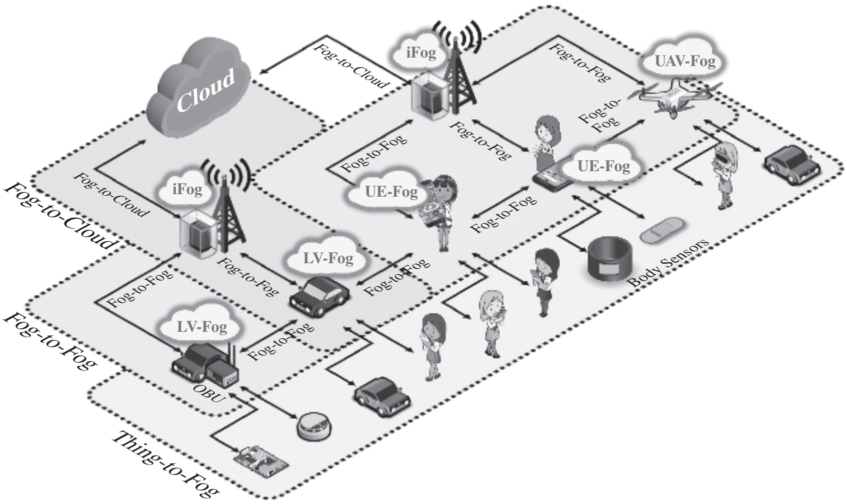

Figure 1.7 The three types of end-to-end networking.

especially when the application requires a certain process or decision making from the cloud.

- *Fog-to-cloud (F2C).* Represents the communication network between the fog node and the cloud, which has a similar influence as F2F to the tenant-side applications. In particular, depending on the underlying infrastructure, geo-location of the cloud and the intercontinental routing path between the fog node and the cloud, the latency between the end-device and the cloud can be very different. Therefore, besides the D2F and F2F, the tenant also needs to consider the F2C network, especially when the application is unable to operate fully in a self-managed manner.

1.5.2 Context-Awareness

The context represents the runtime factors that influence the server operation and the applications. In general, the context of MFC involves the follows:

1.5.2.1 Server Context

Server context influences the serviceability and sustainability of the fog server. Specifically, it involves the following runtime factors:

- *Current task in queue and the task types* [24]. Regardless of the end-to-end communication latency between the tenant-side client and the fog server, the task in queue and the task types are the factors that influence the response time of a tenant-side application deployed on the fog server. In particular, if a fog server has a large number of resource intensive computational tasks in the queue, its response will be much slower than a fog server, which has a decent number of simple tasks in the queue. In addition, the complexity of the task is a relative factor depended on the performance of the fog server.
- *Energy* [62], which is a specific context factor for mobile fog nodes that rely on battery power. For example, UE-fog node and UAV-fog node are commonly operated based on battery power. Hence, tenants need to identify the sustainability of the mobile fog nodes before they decide to deploy the application on them for serving the tenant-side clients.

1.5.2.2 Mobility Context

Mobility context includes a number of movement-related factors that influence the accessibility and connectivity between the tenant-side client and the fog node. In particular, mobility context includes the following elements [27, 62, 63]:

- *Current location and destination,* which represents the current physical geo-location and the destination of the tenant-side client and the fog node, which are the basic parameters to identify the potential encounter rate between the two nodes, based on measuring the possible movement routes.

- *Movement,* which includes the moving frequency (i.e. how often the entity is moving), maximum and average moving speed, acceleration rate (i.e. moving speed of a specific time), moving direction, moving path, and route. Specifically, the system requires continual up-to-date information of these factors in order to adjust the mobility context reasoning.

1.5.2.3 End-to-end Context

The end-to-end context represents the factors that influence the communication performance between the tenant-side client and the fog server. Essentially, a system can measure the end-to-end context based on the server context and mobility context. Hence, one can consider that the end-to-end context is a second level context. In summary, end-to-end context involves the following factors [26, 64]:

- *Signal strength.* A dynamic factor at run-time in many cases, such as LV-fog and UE-fog environments where the density of the wireless network objects is high and hence, they can interfere the signal strength of one another. Therefore, tenants need to consider the signal strength when they intend to measure the connectivity between the fog server and the tenant-side client.
- *Encounter chance and Inter-contact time.* Represent the possibility of the fog server and the tenant-side client to successfully communicate and how long they can maintain the connection for the current stage and next stage. Specifically, it influences the decision of whether the tenants should deploy the application on the fog server for their clients or not, and whether the tenant-side clients can utilize the fog servers for task distribution or not.

1.5.2.4 Application Context

Application context influences the server-side performance and also the tenant-side decision regarding where the application should perform the task. In some cases, the process involves handling large amounts of sensory data locally at the tenant-side client, which could be more energy efficient than distributing the task to a fog server, because the involved data size is too large for the tenant-side client to transmit it to the fog server effectively. In summary, application context involves the following elements [28, 64]:

- *Request data type and the amount of data.* Determines the complexity of the request from the perspective of the computational power of the tenant-side client and the fog server.
- *Deadline of the task and task priority.* Determines the required time to complete the task derived from the tenant-side client. In general, it is a part of the service level agreement (SLA), which specifies that the fog server should adjust the tasks in its queue when it needs to prioritze some tasks.

- *Energy consumption of the client*. Which represents one of the major costs of the tenant-side client, when the tenant-side application demands that the client relay the data to its encountered fog server for processing. As mentioned previously, in some cases local processing at the tenant-side client can be more energy-efficient. Hence, the tenant needs to consider the optimal energy efficiency for the clients in the application management.

1.5.3 Tenant

The nonfunctional requirements of tenants, who are responsible to manage the fog applications, aim to achieve the adaptability of runtime fog applications that involve both application services deployed on the fog servers and the application clients operating on the end-devices. In order to comply with the dynamic MFC environment, tenants need to address the adaptability at each phase of the application management. In general, the management of fog applications, which fundamentally are process management of IoT applications [65], encompasses three basic phases: (re)design, implement/configure and run/adjust.

1.5.3.1 Application Management

(Re)design phase involves the software architecture design and the application process modeling design. Specifically, in MFC, tenants need to consider the adaptivity of their software architecture in terms of self-awareness and deployability. Self-awareness assures that the applications have corresponding mechanisms to identify the situation of the runtime application (e.g. the movement of the end-device or the fog node) and are capable of optimizing the process model automatically with a minimal dependence on the distant central management system. In order to fulfill this requirement, the architecture design may need to support decomposition mechanisms that allow the applications to move the processes of applications or even portions of a process (i.e. tasks) from one node to another node dynamically at runtime.

Implement/configure phase represents the stage that transforms the design abstraction to the executable software and deploying the software to the involved fog nodes and end-devices. In contrast to the classic static IoT systems which do not need to frequently adjust the location of application services, in MFC, based on the runtime context factors of the fog nodes and end-devices, the tenant needs to support the rapid (re)deployment mechanism in order to allow the fog applications to move the processes among the fog nodes toward optimizing the agility of the fog applications. Essentially, the rapid (re)deployment mechanism requires a compatible technical support from the fog infrastructure providers, considering the heterogeneity and dynamic context factors of the fog nodes, the tenants also need to develop the optimal decision-making schemes specifically for their applications in order to deploy the applications to the fog nodes in an optimal manner.

Run/adjust phase represents the runtime application and capabilities of autonomous adjustment based on contextual factors. In general, MFC applications need to support three basic mechanisms and consider two cost factors:

- *Task allocation*. Represents where the tenants should (re)deploy and execute their tasks. In general, unless the entire system utilizes only iFog nodes, MFC applications are rarely operating tasks at fixed locations. Therefore, while the mFog nodes or the tenant-side clients are moving, the fog applications need to continuously determine the next available fog nodes for the clients based on the contextual factors and the specifications (see previous descriptions) in order to rapidly allocate and deploy the tasks to the fog nodes.
- *Task migration*. Has a slight similarity to task allocation in term of (re)deploying tasks at fog nodes. However, task migration can involve much more complexities. For example, in a stream data processing-based fog application, when the client encounters the next fog node while the previous process is in progress, the previous fog node may try to complete the task and then intent to save the process state and wrap the result, the process state information together with the application software (i.e. in case the application is not preinstalled at all the fog nodes) as a task migration package toward sending the task migration to the next encountered fog node of the client. However, there is a chance that the routing path to the new fog node of the client does not exist, which leads to failure of the task migration procedure. Certainly, the example has illustrated only one of the failures in task migration. In order to support the adaptability at the run/adjust phase, the tenants need to consider all the contextual factors and heterogeneity issues in supporting adaptive task migration.
- *Task scheduling*. Represents the timing of any action at the run/adjust phase. In general, based on the application domain, task scheduling can involve different actions including the schedule of task allocation or task migration. For example, in Marine Fog [28], the system needs to identify the best time to route the marine sensory data among the ad hoc network nodes in order to deliver the most important information on time. For example, in LV-Fog, each vehicle needs to perform local measurements in order to identify the encounter and the intercontact time between itself and the incoming vehicle, toward performing rapid information exchange [64].

1.5.3.2 Cost of Energy and Tenancy

Besides the process and task management aspects, tenants need to consider the cost of energy and tenancy. First, *energy cost* commonly refers to the energy consumption of battery-powered end-devices. Here, the tenants should optimize the application design and the runtime processes in order to minimize the energy consumption of the end-devices toward extending the sustainability of the overall processes.

Second, the cost of tenancy indicates the cost-performance trade-off of the application. Specifically, in some cases, the tenants intend to achieve the best agility in terms of task allocation, execution, and migration, they would pre-deploy the application at every single fog node that potentially will be encountered by the end-devices. For example, in the AAL-based application, the tenant might have deployed the application at all the fog nodes on the potential moving path of the patient in order to perform proactive fog-driven sensory data reasoning [18]. However, such an approach may demand a high tenancy cost for the tenant, especially when we consider that the patient (the end-device) may not encounter some of the fog nodes.

1.5.4 Provider

The multi-tenancy-supported fog server providers are responsible to provide adaptive application deployment platforms for the tenants and to cater reliable accessibility to the tenant-side clients.

In order to achieve high QoS for the fog servers, the providers need to address the following aspects.

1.5.4.1 Physical Placement

The physical placement represents where the providers should deploy their fog servers. Commonly, in the case of iFog, the provider may enable fog servers on all the possible nodes (e.g. cellular base stations) and rely on the underlying communication technologies (see Section 1.4) to support the accessibility. On the other hand, in case of mFog [24, 63], providers need to identify the best geo-location to place the mobile fog nodes in order to provide the best QoS to the end-devices and also to support the cost-efficiency of the operation. For example, in UAV-Fog, the provider may choose the locations for the mobile fog nodes based on the density of the end-devices, the signal coverage of the fog node, the distance between the fog server and the end-devices, and the other context factors described in the previous content related to context-awareness. In general, the primary goal of physical placement is to achieve the lowest latency in terms of request/response time, application service handover time, and application task migration time.

1.5.4.2 Server Discoverability and Connectivity

Server discoverability is a specific requirement for the multitenancy fog services, and it involves two phases.

- *Multitenancy fog service provider discovery*. Presents the phase when the tenants intend to discover feasible fog service providers for deploying their applications. Commonly, based on the experience of the cloud service business model, it is likely that the tenants would discover the providers via the indexing services (e.g. Google searching). Alternatively, the providers may establish a federated service registry for the service discovery. Furthermore, the provider may follow the open standard-based service description mechanism or interface (e.g. ETSI – MEC standard) to describe their fog services toward helping the tenants to discover the service that matches to their requirements.
- *Runtime fog server discovery*. Presents the runtime service discovery phase for the fog applications. In general, the fog applications hosted on the fog servers, need to perform seamless interaction with the end-devices on the move. Besides the mobility schemes that help the tenants to identify the movement of the end-devices, tenants need a corresponding mechanism that can help the end-devices to continuously discover and to connect to the new fog servers automatically, without any inference from the end-users. Therefore, the fog servers need to support the corresponding API that allows the tenants to configure the application process/task handover and migration mechanism among the fog nodes. Commonly, if such an API support is not available, tenants have to enable the corresponding mechanisms from the higher layer of the application, which may result in an inefficient tenancy cost and operational performance.

1.5.4.3 Operation Management

In general, fog servers have limited resources in which they can serve a fairly limited number of tenant-side applications at each time slot. Therefore, fog servers require dynamic and optimal mechanisms to support their serviceability. Here, we list the basic elements involved in the operation management of fog servers.

- *Load balancing of request and traffic*. Commonly, fog servers are connecting with one another vertically or horizontally. Therefore, it is possible to establish a cluster computing group among the fog nodes connected in 0-hop range toward enhancing the overall computational capability. Besides the computation-related loads, since fog nodes are fundamentally Internet gateway devices, the heavy network traffic can always affect their serviceability. In order to overcome the traffic-related issues of fog servers, the provider may configure multilayered caching mechanism that utilizes the fog nodes in the hierarchy to reduce the burden [66].

- *Server allocation*, *server scheduling, and server migration*. Three corelated elements, especially for the resource constraint mobile fog node (constraint mFog), such as UE-fog and UAV-fog nodes. To explain, a provider may deploy a specific type of fog server on the constraint mFog device for a domain-specific application in a specific period of time. Whereas, the rest of the time, the device does not operate the fog server at all. For example, in an indie fog environment [17], the owner of a smartphone (UE)-fog may configure the device to serve the context reasoning–based fog server [36] only when the owner is carrying the device in outdoor areas and the battery level of the device is over 50%. Further, the owner can also configure that, when the battery level of the device is between 51 and 70%, it will redirect/migrate the request to another authorized fog node. Similarly, the notion described here is applicable to other MFC domains, such as LV-fog [24] and UAV-fog [63].

1.5.4.4 Operation Cost

Operating fog servers can be costly for the providers especially when the providers are unable to identify what tenants really demand. For example, stream data filtering is a common method used in fog computing and the corresponding program can be quite simple in comparison to the scientific programs operated on the cloud. However, in the classic approach, the tenant may need to wrap the simple program to a package that runs on the resource-intensive VM environment because the provider was following the classic cloud service deployment approach to providing the fog server. Explicitly, the provider has inefficiently increased the burden of the fog node while providing excess service to tenants who demanded only a simple method. Therefore, the provider should consider what are the most cost-efficient service types that should be supported by the fog servers. Beside the service type, the providers of the battery-powered mobile fog nodes need to specifically address the energy-efficient of the fog servers in order to improve their sustainability. For instance, although providers can easily replace UAV-Fog nodes, considering the extra latency derived from the process/task handover and migration while replacing the UAV-Fog nodes, frequently replacing UAV-Fog nodes will reduce the QoE for the tenant-side clients.

1.5.5 Security

In large-scale MFC systems, the classic perimeter-based security approach will not suffice, security strategies in MFC must account for various factors: physical, end-to-end, and also monitoring and management.

1.5.5.1 Physical Security

Since the fog nodes will be deployed in the wild (e.g. road-side infrastructure in LV-Fog), physical exposure is a more serious threat than in conventional enterprise or cloud computing. The devices need antitamper mechanisms that prevent, detect, and respond to intrusions, while simultaneously considering how to allow maintenance operations without compromising these mechanisms [61].

1.5.5.2 End-to-End Security

End-to-end security is concerned with the security capabilities of each device within the MFC, spanning different layers of the fog architecture and devices therein.

- *Execution environments*. The devices need to include capable software and hardware components solely dedicated to performing security functions (so-called roots of trust (RoT). These components should, on one hand, be isolated from the rest of the platform while also verifying the functions performed by the platform.

  Based on RoT-s, the nodes must have the capability to provide trusted execution environments. In the case of virtualized environments, this can be achieved through virtual trusted platform modules.
- *Network security*. According to the OpenFog security requirements, fog nodes should provide the security services defined by the ITU X.800 recommendation by using standard-based secure transport protocols.

  Some nodes in the MFC system can provide security services on the network through network function virtualization (NFV) and SDN, for example, deep packet inspection.
- *Data Security* protection of data must be taken care of in all the mediums in which data may lie or move: in system memory, in persistent storage or data exchanged over the network.

1.5.5.3 Security Monitoring and Management

The system must be capable of observing security state in the network and reacting to new threats via monitoring and management mechanisms. Security management should allow definition and updating of security policies and propagation of the policies over the network in real-time.

In addition to policy management, identity and credential management is a requirement. In addition to the necessary registration and credential storage functions, an MFC system must handle the challenge of authentication and access control in situations with intermittent connectivity, e.g. negotiating session keys when crossing different trust domains while ensuring data integrity and privacy [67]. Security monitoring, on the other hand, should collect log traces while ensuring their integrity.

The OpenFog security requirements define at least two logically separate security domains: (1) policies concerning the collection of Fog entities within the system that can interact with one another and (2) for policies regarding the individual services and applications being executed and provided on the platform.

1.5.5.4 Trust Management and Multitenancy Security

Managing trust. The information flows in MFC raise the issue of how to determine which nodes in the network are trustworthy for a particular client and request? Trust management frameworks need to manage trust assessment of both devices and applications and be able to adjust to the real-time updates of trust-related data, such as social relationships and execution results [68]. Some scenarios may also require decentralized trust management achievable with the help of blockchain technology; however, this aspect has issues with latency [69].

Multitenancy. As the fog architecture dictates that fog platforms can host services for multiple parties, this poses the issue of how to ensure isolation in the runtime environment and ensure that only data that was intended to be shared is available across the instances [61, 70]. Secure isolation of tenant and user space continues to be a challenging requirement, as vulnerabilities allowing adversaries to access memory of other tenants VMs without permission [71] have surfaced, even in mass-produced CPUs.

1.6 Open Challenges

Fog computing has been introduced to overcome many challenges that cloud computing was incapable of handling, such as latency-sensitivity, connectivity between the large set of cloud-based applications, etc. Therefore, many researchers investigated and proposed different architectures for introducing fog computing into the traditional cloud computing to create an extension of the existing design. This action resolved into solving some of the old obstacles and generate new perspectives and ambitions that led to new challenges.

1.6.1 Challenges in Land Vehicular Fog Computing

Introducing the cloud computing into VANET was redemption since it provided a solution to most challenges that traditional design of VANET faced, such as decreased flexibility, scalability, poor connectivity, and inadequate intelligence [72]. However, the new generation of VANET has introduced new requirement with respect to the high mobility, low latency, real-time applications, and reliable connectivity. For all these reasons, adding fog computing to the equation has emerged as a potential solution. Based on the existing research it seems that there are still some obstacles to be dealt with, for example, the management of fog server in geographically distributed fog nodes. The difficulty relies on the positioning of each edge server in a given area, which is important for fog approach and it considers many parameters, such as vehicles density, traffic status data, and processing load on the servers. Another challenge is related to management of neighboring fog servers with respect to communication and access, which can be affected by the environment. Finally, the assessment and handling dissemination of real-time critical message can be problematic, since design should be capable of distinguishing between a true or false event [73].

1.6.2 Challenges in Marine Fog Computing

Existing works [4, 28] have proposed the frameworks for improving the efficiency of the communication and for application management in the hybrid MFC environment that consist of both iFog and mFog nodes. Essentially, comparing to the UAV-Fog nodes and the UE-fog nodes, the Marine Fog nodes do not have the resource-constraint issue in terms of computation and data storage. However, because the operating marine environment lacks infrastructure, the communication between the Marine Fog nodes and the distant central cloud faces the latency issue derived from the limitation of the underline network topology and technology. For example, the bandwidth of the common VHF radio used in maritime communication can reach only 28.8 kbps and the range of the infrastructure-less 4G/5G LTE device-to-device communication is unable to fulfill the need of a marine environment. In order to overcome the fundamental communication issue, developers may consider integrating UAV-fog nodes [31] in which the system can deploy the UAV-fog nodes between vessels to form a mesh network toward dynamically supporting better bandwidth and more stable connection. However, UAVs have a limited available operation time slot because they are battery-powered. The system needs to introduce an adaptive scheduling scheme, physical location placement scheme. Further, the system needs to dynamically adjust the movement of the UAV-fog nodes based on the interconnected vessels in order to seamlessly maintain the mesh network.

1.6.3 Challenges in Unmanned Aerial Vehicular Fog Computing

Current works in UAV mFog [32, 52] were focusing on the underlying system design and communication mechanisms. Although UAV-Fog nodes have many potential applications due to the features in terms of fast deployment, scalability, flexibility, and cost-efficiency [31], integrating UAV-Fog nodes to pervasive computing systems or IoT systems raises many new challenges besides the underlying communication mechanisms. Specifically, existing works have not fully addressed the requirements for both tenant and provider. For example, although an existing work [63] has proposed schemes that enable UAV-Fog nodes to perform data-driven service handover, which transfers the client data from one UAV-Fog node to another, this scheme was designed for a domain-specific application in which the author assumes the system has preinstalled the application to all the UAV-Fog nodes and hence, at runtime, the UAV-Fog nodes need only to transfer the client data in order to support the mobility. On the other hand, considering the multitenancy fog service model, preinstalling applications to the UAV-Fog nodes for all the tenants will cause a high burden to the storage size, especially when the application involves large size files. Therefore, UAV-fog nodes require the mechanism that supports rapid and dynamic application management in terms of task allocation/placement and task migration.

1.6.4 Challenges in User Equipment-based Fog Computing

There exist a large number of frameworks designed for supporting mobile UE from iFog. However, existing works rarely address the challenges in UE-fog nodes. The use cases described in the previous section indicate that systems which integrate UE-fog nodes require a dynamic program deployment mechanism. For example, in the advanced crowd sensing use case, which utilize UE-fog nodes to provide the interpreted context information derived from the collected sensory data, the UE-fog nodes need to provide the corresponding service that allows the clients to deploy the program code of the context reasoning algorithm on the UE-fog nodes. Explicitly, considering the UE-fog nodes have constraint resources, they are unable to support the common VM or containers engine-based service for the dynamic program deployment. Instead, developers generally would develop the standalone solutions which leave the interoperability as an unsolved problem in UE-fog. In order to address such an issue, developers may consider integrating an open standard–based service interface or to develop a specific mobile fog node description language based on the extension of existing cloud service-based standard, such as OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA).

1.6.5 General Challenges

1.6.5.1 Testbed Tool

The complexity of the MFC topology, which encompasses the hierarchical, vertical, and horizontal interconnections among the cloud, the iFog, the mFog, and the end-devices, has increased the challenges in system design and validation. Commonly, the developers of the distributed computing system, such as cloud services, pervasive services, or mobile services, have a broad range of simulation options (e.g. CloudSim [http://www.cloudbus.org/cloudsim], the ONE simulator [https://www.netlab.tkk.fi/tutkimus/dtn/theone], and iFogSim [https://github.com/Cloudslab/iFogSim], etc.) to validate their system designs. However, existing simulation tools are insufficient to validate many MFC systems because they are unable to address all the elements of MFC. For example, although the cloud service–based simulation tools are capable of simulating the hardware heterogeneity, they do not include mobility-related factors. For another example, while the mobile service-based simulation tools are capable of simulating the movement of entities, they do not have corresponding mechanisms to simulate the complete MFC network that contains the hierarchical and the vertical interconnection among the entities. Finally, iFogSim is capable of simulating the stationary fog nodes but it does not provide the mechanism to simulate the mobile fog nodes. Consequently, integrating the existing tools to develop a comprehensive MFC testbed becomes a critical challenge.

1.6.5.2 Autonomous Runtime Adjustment and Rapid Redeployment

To achieve optimal operation in MFC, the system demands autonomous adjustment and rapid redeployment based on context-awareness and the real-time system process analysis. In particular, considering an MFC system with the large-scale deployment of mFog and iFog nodes, manned optimization becomes impractical and inefficient. In order to overcome such an issue, the system needs to introduce a certain level of self-aware mechanisms to the fog nodes. Specifically, at an early stage, the system manager can preconfigure the basic knowledge to the fog nodes that help the fog nodes to identify the situation at runtime and to adjust or to redeploy the fog service. While the system continuously operates, the fog nodes should support edge intelligence mechanism in which the fog nodes together with the back-end cloud can study the historical records of the operation in order to identify the weak parts and to perform adjustment and redeployment automatically. For example, by enabling edge intelligence on the UAV-Fog nodes, the UAV-Fog nodes are capable of learning when and where to adjust their location, when and where they should migrate or redeploy their services, or when they should reserve their computational resources in order to provide the best QoE to the tenant-side clients.

1.6.5.3 Scheduling of Fog Applications

A few works have addressed fog application scheduling for hybrid MFC environments that consist of both iFog and mFog nodes. However, existing frameworks for mFog either designed for a specific purpose, such as for data routing in SDN [28], or for process/task distribution [20, 64], have not considered all the contextual factors (see Section 1.5.2) and the heterogeneity factors (see Section 1.5.1). Specifically, besides the movement of the entities, the tenants' application scheduling scheme should consider that the fog servers have the different computational and networking performance at different time slot due to the hardware specifications and the runtime context factors. In other words, developers need to have insight into the processing delay by considering all the factors toward proposing the optimal application scheduling scheme.

1.6.5.4 Scalable Resource Management of Fog Providers

In general, fog nodes have limited resources to serve tenants because they are fundamentally the independent network gateway devices that do not interconnect with one another in a short range like the server pools in the cloud. In other words, introducing computational scalability in MFC faces the network latency challenge. Commonly, providers of fog servers may manage multiple fog servers that are interconnected vertically within the hierarchy or are interconnected horizontally in a peer-to-peer manner. However, since the primary objective of fog servers in MFC is to serve the tenant-side clients, the distances between the fog servers are rarely within the range that is capable of achieving ultra-low latency. Therefore, the classic cloud-based scalability scheme is incompatible in MFC and hence, scalability becomes an unsolved challenge, especially for mFog environments. In order to address the challenge in scalability, the developers may consider developing a hybrid framework that integrates opportunistic computing, SDN, and context-aware software architecture toward enabling an adaptive fog service topology that can be orchestrating the fog servers in a highly dynamic manner.

1.7 Conclusion

Fog computing has appeared as a paradigm that extends Cloud computing and offers interesting and promising possibilities to overcome the limitations of the traditional environment. However, fog computing still faces a challenge with respect to mobility when the tasks come from ubiquitous mobile devices or applications in which the data sources are in constant movement. Therefore, this chapter pointins a spotlight on the development and advancement in the MFC and its related models. In this process, it reflects the areas where MFC is needed and has a very positive impact on enhancing the existing systems and opening new directions for evolving the environment, such as in the infrastructures, equipment and devices, land and marine vehicles, autonomous vehicles, etc. In addition, it investigates the communication technologies that have emerged or upgraded based on MFC with an emphasis on the added value that facilitated the process of solving some of the big challenges existing in the traditional design. Furthermore, the chapter also presents, in a comprehensive manner, the basics of nonfunctional requirement needed to achieve the basic QoS principles. Finally, it addresses the important open challenges that still need to be addressed in the new environment designed under the MFC framework.

Acknowledgment

The work is supported by the Estonian Centre of Excellence in IT (EXCITE), funded by the European Regional Development Fund.

References

1 Chang, C., Srirama, S.N., and Buyya, R. (2019). *Internet of Things (IoT) and New Computing Paradigms*, ch. 1, p. 1. Wiley.

2 Alam, K.M., Saini, M., and El Saddik, A. (2015). Toward social iInternet of vehicles: concept, architecture, and applications. *IEEE Access* 3: 343–357.

3 Gharibi, M., Boutaba, R., and Waslander, S.L. (2016). Internet of drones. *IEEE Access* 4: 1148–1162.

4 Al-Zaidi, R., Woods, J.C., Al-Khalidi, M., and Hu, H. (2018). Building novel VHF-based wireless sensor networks for the Internet of marine things. *IEEE Sensors Journal* 18 (5): 2131–2144.

5 Santagati, G.E. and Melodia, T. (2017). An implantable low-power ultrasonic platform for the Internet of medical things. In: *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, –1, 9. IEEE.

6 Chang, C., Hadachi, A., Srirama, S.N., and Min, M. (2018). *Mobile Big Data: Foundations, State of the Art, and Future Directions*, 1–12. Cham: Springer International Publishing.

7 Chang, C., Srirama, S.N., and Mass, J. (2015). A middleware for discovering proximity-based service-oriented industrial Internet of Things. In: *2015 IEEE International Conference on Services Computing (SCC)*, 130–137. IEEE.

8 Satyanarayanan, M., Bahl, P., Cáceres, R., and Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing* 4: 14–23.

9 ETSI (2017). *ETSI and OpenFog Consortium collaborate on fog and edge applications*. Berlin, Germany: MEC Congress https://www.etsi.org/newsroom/news/1216-2017-09-news-etsi-and-openfog-consortium-collaborate-on-fog-and-edge-applications.

10 M. Iorga, L. Feldman, R. Barton et al., Fog Computing Conceptual Model: Recommendations of the National Institute of Standards and Technology, Special Publication NIST SP 500-325, National Institute of Standards and Technology, March 2018.

11 Satyanarayanan, M. (2017). The emergence of edge computing. *Computer* 50 (1): 30–39.

12 Fahmy, H.M., El Ghany, M.A., and Baumann, G. (2018). Vehicle risk assessment and control for lane-keeping and collision avoidance at low-speed and high-speed scenarios. *IEEE Transactions on Vehicular Technology* 57 (6): 4805–4818.

13 Hafner, M.R., Cunningham, D., Caminiti, L., and Del Vecchio, D. (2013). Cooperative collision avoidance at intersections: algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems* 14 (3): 1162–1175.

14 Nguyen, V., Kim, O.T.T., Pham, C. et al. (2018). A survey on adaptive multichannel mac protocols in vanets using markov models. *IEEE Access* 6: 16493–16514.

15 Vinel, A., Lyamin, N., and Isachenkov, P. (2018). Modeling of V2V communications for C-ITS safety applications: a CPS perspective. *IEEE Communications Letters* 22 (8): 1600–1603.

16 Kalatzis, N., Avgeris, M., Dechouniotis, D. et al. (2018). Edge computing in IoT ecosystems for UAV-enabled early fire detection. In: *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 106–114. IEEE.

17 Chang, C., Srirama, S.N., and Buyya, R. (2017). Indie fog: an efficient fog-computing infrastructure for the Internet of Things. *Computer* 50 (9): 92–98.

18 Soo, S., Chang, C., Loke, S.W., and Srirama, S.N. (2017). Proactive mobile fog computing using work stealing: data processing at the edge. *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)* 8 (4): 1–19.

19 Soto, V., De Grande, R.E., and Boukerche, A. (2017). REPRO: time-constrained data retrieval for edge offloading in vehicular clouds. In: *Proceedings of the 14th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 47–54. ACM.

20 Zhu, C., Tao, J., Pastor, G. et al. (2018). Folo: latency and quality optimized task allocation in vehicular fog computing. *IEEE Internet of Things Journal* 6 (3): 4150–4161.

21 Yao, H., Bai, C., Zeng, D. et al. (2015). Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurrency and Computation: Practice and Experience* 27 (18): 5780–5792.

22 Cordeschi, N., Amendola, D., and Baccarelli, E. (2015). Reliable adaptive resource management for cognitive cloud vehicular networks. *IEEE Transactions on Vehicular Technology* 64 (6): 2528–2537.

23 Chen, Y.-A., Walters, J.P., and Crago, S.P. (2017). Load balancing for minimizing deadline misses and total runtime for connected car systems in fog computing. In: *2017 IEEE International Symposium on Parallel and Distributed*

Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), 683–690. IEEE.

24 Fan, X., He, X., Puthal, D. et al. (2018). CTOM: collaborative task offloading mechanism for mobile cloudlet networks. In: *2018 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.

25 Marshall, W.E. and Dumbaugh, E. (2018). Revisiting the relationship between traffic congestion and the economy: a longitudinal examination of us metropolitan areas. *Transportation*: 1–40.

26 Wang, C., Li, Y., Jin, D., and Chen, S. (2016). On the serviceability of mobile vehicular cloudlets in a large-scale urban environment. *IEEE Transactions on Intelligent Transportation Systems* 17 (10): 2960–2970.

27 Wang, Z., Zhong, Z., Zhao, D., and Ni, M. (2018). Vehicle-based cloudlet relaying for mobile computation offloading. *IEEE Transactions on Vehicular Technology* 67 (11): 11181–11191.

28 Yang, T., Cui, Z., Wang, R. et al. (2018). A multivessels cooperation scheduling for networked maritime fog-ran architecture leveraging SDN. *Peer-to-Peer Networking and Applications* 11 (4): 808–820.

29 R. Sosa, R. Sucasas, A. Queralt et al., Towards an open, secure, decentralized and coordinated fog-to-cloud management ecosystem, D5.1 mF2C reference architecture (integration IT-1), mF2C Consortium, 2018.

30 Xu, G., Shen, W., and Wang, X. (2014). Applications of wireless sensor networks in marine environment monitoring: a survey. *Sensors* 14 (9): 16932–16954.

31 Mohamed, N., Al-Jaroodi, J., Jawhar, I. et al. (2017). UAV fog: a UAV-based fog computing for Internet of Things. In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 1–8. IEEE.

32 Radu, D., Cretu, A., Parrein, B. et al. (2018). Flying ad hoc network for emergency applications connected to a fog system. In: *International Conference on Emerging Internetworking, Data & Web Technologies*, 675–686. Springer.

33 451 Research, "Size and impact of fog computing market," OpenFog Consortium, 2017.

34 Puliafito, C., Mingozzi, E., and Anastasi, G. (2017). Fog computing for the Internet of mobile things: issues and challenges. In: *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 1–6. IEEE.

35 Silva, P.M.P., Rodrigues, J., Silva, J. et al. (2017). Using edge-clouds to reduce load on traditional Wi-Fi infrastructures and improve quality of experience. In: *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, 61–67. IEEE.

36 Chang, C. and Srirama, S.N. (2018). Providing context as a service using service-oriented mobile indie fog and opportunistic computing. In: *European Conference on Software Architecture*, –219, 235. Springer.

37 Siddiqui, F., Zeadally, S., and Salah, K. (2015). Gigabit wireless networking with ieee 802.11 ac: technical overview and challenges. *Journal of Networks* 10 (3): 164.

38 Rejiba, Z., Masip-Bruin, X., Jurnet, A. et al. (2018). F2C-aware: enabling discovery in Wi-Fi-powered fog-to-cloud (F2C) systems. In: *2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 113–116. IEEE.

39 Chowdhury, M., Steinbach, E., Kellerer, W., and Maier, M. (2018). Context-aware task migration for HART-centric collaboration over FiWi based tactile internet infrastructures. *IEEE Transactions on Parallel and Distributed Systems* 29 (6): 1231–1246.

40 Enayet, A., Razzaque, M.A., Hassan, M.M. et al. (2018). A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities. *IEEE Communications Magazine* 56 (2): 110–117.

41 Taleb, T., Dutta, S., Ksentini, A. et al. (2017). Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine* 55: 38–43.

42 Akter, M., Zohra, F.T., and Das, A.K. (2017). Q-MAC: QoS and mobility aware optimal resource allocation for dynamic application offloading in mobile cloud computing. In: *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 803–808. IEEE.

43 Lei, L. (2016). Stochastic modeling of device-to-device communications for intelligent transportation systems. In: *2016 23rd International Conference on Telecommunications (ICT)*, 1–5. IEEE.

44 ITUR, Requirements related to technical performance for IMT-advanced radio interface(s), Report M.2134, International Telecommunications Union, 2008.

45 ITUR, Minimum requirements related to technical performance for IMT 2020 radio interface(s), Report M.2410, International Telecommunications Union, 2017 (22nd February Draft).

46 IMT vision – framework and overall objectives of the future development of IMT for 2020 and beyond, Recommendation ITU-R M.2083-0, pp. 2083–2090, International Telecommunications Union, 2015.

47 Dong, P., Zheng, T., Yu, S. et al. (2017). Enhancing vehicular communication using 5g-enabled smart collaborative networking. *IEEE Wireless Communications* 24: 72–79.

48 Yu, F., Chen, H., and Xu, J. (2018). DMPO: dynamic mobility-aware partial offloading in mobile edge computing. *Future Generation Computer Systems* 89: 722–735.

49 Wang, Z., Zhao, Z., Min, G. et al. (2018). User mobility aware task assignment for mobile edge computing. *Future Generation Computer Systems* 85: 1–8.

50 Nasrin, W. and Xie, J. (2018). SharedMEC: sharing clouds to support user mobility in mobile edge computing. In: *2018 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.

51 Yang, J., Wen, J., Jiang, B. et al. (2018). Marine depth mapping algorithm based on the edge computing in Internet of Things. *Journal of Parallel and Distributed Computing* 114: 95–103.

52 Jeong, S., Simeone, O., and Kang, J. (2018). Mobile edge computing via a UAV-mounted cloudlet: optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology* 67 (3): 2049–2063.

53 Salem, A., Salonidis, T., Desai, N., and Nadeem, T. (2017). Kinaara: distributed discovery and allocation of mobile edge resources. In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 153–161. IEEE.

54 Liyanage, M., Chang, C., and Srirama, S.N. (2018). Adaptive mobile web server framework for mist computing in the Internet of Things. *International Journal of Pervasive Computing and Communications.*

55 Sucipto, K., Chatzopoulos, D., Kosta±, S., and Hui, P. (2017). Keep your nice friends close, but your rich friends closer — computation offloading using nfc. In: *IEEE INFOCOM 2017 IEEE Conference on Computer Communications*, 1–9.

56 Characteristics of VHF radio systems and equipment for the exchange of data and electronic mail in the maritime mobile service RR appendix 18 channels, Recommendation M.1842-1, International Telecommunication Union, 2008.

57 Al-Zaidi, R., Woods, J., Al-Khalidi, M. et al. (2017). Next generation marine data networks in an iot environment. In: *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 50–55. IEEE.

58 Manoufali, M., Alshaer, H., Kong, P.-Y., and Jimaa, S. (2013). Technologies and networks supporting maritime wireless mesh communications. In: *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, 1–8. IEEE.

59 Bardram, A.V.T., Larsen, M.D., Malarski, K.M. et al. (2018). Lorawan capacity simulation and field test in a harbour environment. In: *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 193–198.

60 Raza, U., Kulkarni, P., and Sooriyabandara, M. (2017). Low power wide area networks: an overview. *IEEE Communication Surveys and Tutorials* 19: 855–873.

61 Martin, B.A., Michaud, F., Banks, D. et al. (2017). Openfog security requirements and approaches. In: *2017 IEEE Fog World Congress (FWC)*, 1–6.

62 Sui, Y., Wang, X., Pengt, M., and An, N. (2017). Optimizing mobility and energy charging for mobile cloudlet. In: *2017 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.

63 Tang, F., Fadlullah, Z.M., Mao, B. et al. (2018). On a novel adaptive UAV-mounted cloudlet-aided recommendation system for LBSNs. *IEEE Transactions on Emerging Topics in Computing* 7 (4): 565–577.

64 Truong-Huu, T., Tham, C.-K., and Niyato, D. (2014). To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud. In: *2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)*, 182–189. IEEE.

65 Chang, C., Srirama, S.N., and Buyya, R. (2016). Mobile cloud business process management system for the Internet of Things: a survey. *ACM Computing Surveys* 49, pp. 70:1–70:42.

66 Li, M., Yu, F.R., Si, P. et al. (2018). Software-defined vehicular networks with caching and computing for delay-tolerant data traffic. In: *2018 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.

67 Kumar, N., Rodrigues, J.J.P.C., Guizani, M. et al. (2018). Achieving energy efficiency and sustainability in edge/fog deployment. *IEEE Communications Magazine* 56: 20–21.

68 Ruan, Y., Durresi, A., and Uslu, S. (2018). Trust assessment for Internet of Things in multiaccess edge computing. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, –1155, 1161. IEEE.

69 Wang, S., Xu, J., Zhang, N., and Liu, Y. (2018). A survey on service migration in mobile edge computing. *IEEE Access* 6: 23511–23528.

70 Zhang, P., Zhou, M., and Fortino, G. (2018). Security and trust issues in fog computing: a survey. *Future Generation Computer Systems* 88: 16–27.

71 Lipp, M., Schwarz, M., Gruss, D. et al. (2018). Meltdown: reading kernel memory from user space. In: *27th Security Symposium*, 973–990.

72 Hussain, R., Son, J., Eun, H. et al. (2012). Rethinking vehicular communications: merging vanet with cloud computing. In: *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, 606–609. IEEE.

73 Nobre, J.C., de Souza, A.M., Rosário, D. et al. (2019). Vehicular software-defined networking and fog computing: integration and design principles. *Ad Hoc Networks* 82: 172–181.