Mobile Edge Computing: Architecture, Use-cases, Applications

Craig Pritchard, Yousef Beheshti, Mohammad Sepahi

Abstract— By enormous growth in IoT and smart devices and the advent of many new applications, Internet traffic volume has been growing exponentially. Analyzing such flooding traffic requires enormous compute and bandwidth and raises privacy concerns. Edge platforms can become the tool to ease the burden by bringing resources to the proximity of data. Therefore, new architectures, which bring network functions and contents to the network edge, are proposed, i.e., mobile edge computing and caching. In this survey, we make an exhaustive review on the literature research efforts on mobile edge networks. We give an overview of mobile edge networks, including definition, architecture, and application and use-cases . We then survey the issues related to computing, caching, and communication techniques at the network edge with the focus on applications and use cases of mobile edge networks.

I Introduction

Many mobile applications relay on remote data centers. This imposes high loads of data in mobile networks due to uploading and downloading data to and from data centers. Demand of bandwidth is expected to be doubled each year [1]. Moreover, mobile devices have the computation power of a sever in a decade ago. With the increase of computational power, novel mobile applications such as *Augmented Reality (AR)* become realistic.

On the other hand, Internet of Things (IoT) enables resource-limited devices to interconnect with the Internet. To support IoT, new techniques such as *computation offloading* are introduced in order to offload a part of the computation to a remote cloud server. Although the offloading brings lower energy consumption and computation power, it may introduce extra latency while exchanging data between the device and cloud servers. To address this issue, cloudlet offloading has become prevalent. In this technique the computational task is offloaded to a cloud server in the proximity of the mobile device using Wi-Fi [2]. However, Cloudlet has its own pitfalls. First it is only accessible through Wi-Fi that will only cover a short range, Second, it is not scalable in terms of resource provisioning.

To mitigate all the limitations and issues mentioned above, a new paradigm called Mobile Edge Computing (MEC) has been introduced. This concept was firstly proposed by the European Telecommunications Standard Institute (ETSI) in 2014, and was defined as follows : *"provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers"* [3].

MEC aims to reduce latency by bringing the computation and storage capacity from the core network to the edge network. The main promises of MEC are low latency, high bandwidth, and real-time availability of radio network information that can be leveraged by applications and lead to higher QoE for the end-users. Many mobile applications can benefit from MEC by offloading their computational tasks to the edge servers. For instance, providing realtime network information (e.g., network load, user's location information) enables developing context-aware applications. The term *"Edge"* may refer to both the base stations such as eNodeB, Radio Network Controller, and data centers close to the radio network [4]. MEC is implemented based on Network function virtualization (NFV) that a single edge device can provide computation power for multiple devices by creating multiple Virtual Machines (VMs) in order to perform different tasks simultaneously [5]. There are other surveys that investigate edge computing from communication perspective[6] and convergence of mobile edge computing and deep learning [7]. However, these topics are out of the scope of this survey.

Mobile Edge Computing The Table 1 shows the major differences between Mobile Cloud Computing (MCC) and MEC in different [6]

Table 1: My caption

	MEC	**MCC**
Server Hardware	Small-scale data centers	Large-scale data centers
Server Location	Co-located with wireless gateways, WiFi routers, LTE BSs	Installed at dedicated buildings, with size of several football fields
Distance to End -User	Small (tens to hundreds of meters)	Large (may across continents)
System Management	Hierarchical control (centralized/distributed)	Centralized control
Supportable latency	Less than tens of milliseconds	Larger than 100 milliseconds
Applications	Computation intensive, Latency sensitive (AR, surveillance system)	Latency-tolerant and computation-intensive applications , automatic driving, social networking, interactive online gaming, mobile commerce/health/learning

There are many areas in the young field of Mobile Edge Computing (MEC) including System models, architectures, enabling techniques, applications, edge caching, edge computation offloading, and connections with IoT and 5G. the contribution of this paper is two-fold: (a) Providing a survey covering important works that are conducted in the literature (b) Discussing existing challenges in the architecture, deployment, and standardization of MEC along with possible future research directions in this field.

The organization of this paper is as follows: Section II provides the motivation behind MEC by describing different characteristics, application and use cases that are introduced by other researches. Section III summarizes the standardized MEC framework identified by ISG. key use cases of IoT in MEC are discussed in Section IV.

II Motivation

Here are list of scenarios where MEC can be applied. This list is collectively derived from prior works on MEC [8, 6]. Figure 1 summarizes applications and use cases introduced by the state-of-the-art in MEC.

A Dynamic Content Optimization

Content optimization is usually done at hosting site where it uses user history [9]. If the content optimizer is hosted at edge server, it can uses accurate information such as the current user location, network load, network status, etc,. Leveraging edge based content optimizer can lead to higher quality of experience for the end user.

B Computational Offloading in IoT

Compute-intensive applications such as surveillance systems need a lot of computational. On the other hand, there is a convergence of such systems with IoT devices in which computational power is restricted. In this scenario usually tasks are split among the IoT devices, and the intensive part is transferred to the core network e.g,. cloud. However, this case can be optimized by bringing the intensive part to edge rather than transferring to the cloud. Offloading techniques are comprehensively are surveyed in [10].

C Surveillance and Video analytics

Authors in [11] propose a system that demonstrate that real time video analytics is the killer application for the edge computing. video analytics for edge computing applications range from [12] to smart transportation[13]. IoT devices need to cope with compute-intensive applications. With dropping camera prices and increasing accuracy of deep neural networks (DNNs), we see an explosive growth of video-analytics applications [14, 15] and deployments of large camera networks [16].If the computational offloading is done at edge, it can reduce the energy consumption in IoT devices. Experimental results in *eyeDentify* [17] shows significant energy saving by proper computational offloading. Authors in [18] propose a system built on top of an edge computing platform, which offloads computation between clients and edge nodes, collaborates nearby edge nodes, to provide low-latency video analytics at places closer to the users. Recently, There in a new trend on collaborative video analytics at the edge, authors in [19, 20, 21] are the first to provide new insights and potential directions that can benefit more from collaborative notion of edge computing.

D Mobile Big Data Analytics

Structured and non-structured data form Big Data. These data are analyzed in order to enable companies for better business decisions. To process this data, it is transformed from edge devices to the core network [22]. Due to high volume of the data, this analytics require high bandwidth and face latency. In such a scenario, MEC can play a role by bringing the computation to the edge, and transferring the results back to the core. One more application of edge can be Ocean monitoring, where ocean climate and changes are monitored to discover possible disasters such as tsunami. The sensors in oceans provide large amount of data that need high bandwidth and computing power. Due to wide area network latency, this process face additional latency. To this end, edge servers can be deployed to reduce this latency [23].

E Smart Transportation

Nowadays, cities are facing many challenges such as lack of public transportation, limited number of parking, pedestrian and driver's safety [24]. To address these challenges IoT devices and collect real-time data. However, in this scenario this data should be processed with low latency in order to provide real-time result that will lead to better decisions for the end applications. For instance, a driver can be signaled about a pedestrian that is crossing the street, or about traffic jams in certain routes.

F Enhancing Security and Privacy

Cloud computing servers can be target of attacks. Moreover, they do not provide privacy for the end users. An enterprise can benefit from proximity edge servers to ensure privacy for the employees. There is no need for each user to interact with cloud individually. for a task that needs user's location information, this information can be sent to edge (edge can be a cloudlet that the enterprise can benefit from it). When the aggregation and computation is done, edge server can send this data to cloud.

III MEC Framework

ISG has defined the framework and reference architecture [26] for MEC . The entities are categorized into three groups: System level, host level, and network level.

the host level consists of Mobile host edge and host level management entities. Mobile host edge entity includes: ME platform, ME application, and infrastructure. It is a host for storage and network resources for applications.

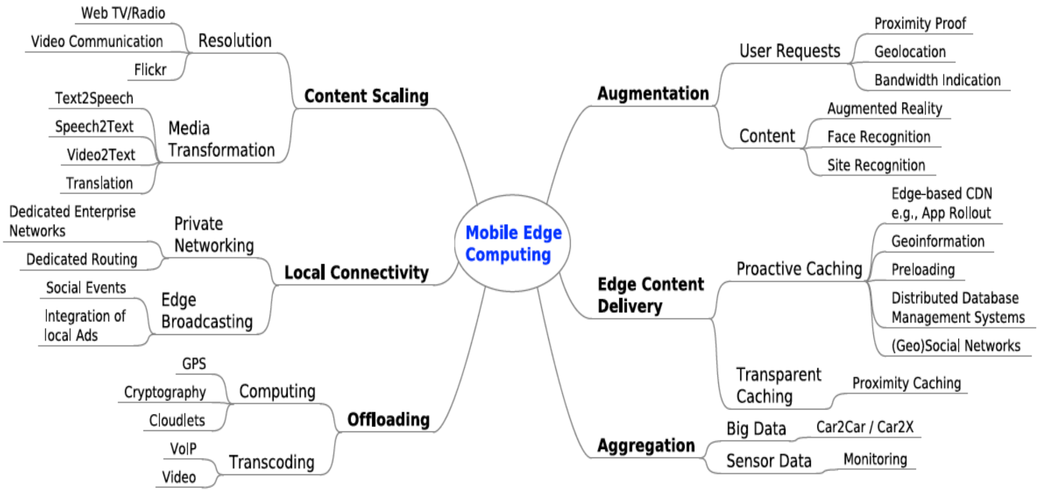

Figure 1: Mobile Edge Computing: Applications and Use-Cases [25]

The networks level covers entities such as 3GPP cellular network, local networks and external networks. This level illustrates the connectivity to local area networks, cellular networks and external networks. The system level management has the overall visibility to the system.

IV MEC and IoT

In this section we first introduce key use cases for IoT and MEC which have been identified by ETSI MEC ISG [26]. Later, we show how MEC can be used for these cases.

A Use Cases of IoT and MEC

B Vehicle-to-Infrastructure communication

It is anticipated that the fist self-driving cars are commercially available by 2020. The most of the use cases of this technology are related to connected cars. IoT is the key element of this kind of communication and plays an important role for such technologies. Use cases for connected cars are not only about self-driving, but also smart transportation such as road safety services. There are also works that design communication protocols for vehicular and beyond devices such as drones by machine learning techniques [27], [28], [29], [30], and [31].

C Computation offload into the edge cloud

Part of the computation part of applications running on mobile devices can be offloaded to cloud. Such offloading is strictly useful for IoT devices where the end terminal is limited in terms of power and requires to prolong battery life time. Moreover, MEC can provide low latency for the applications. Zhang et al [32] propose a hierarchical cloud-based Vehicular Edge Computing (VEC) offloading framework, where a backup computing server in the neighborhood is introduced to make up for the deficit computing resources of MEC servers.

D MEC Deployment in IoT based on Mobility

Prior work [4] depicts two types of deployments for MEC based on the fixed and mobile IoT devices. In the following the two deployments are described with corresponding use case examples

D.1 IoT devices are Fixed

As mentioned in the Section II, one of the application of MEC is in surveillance systems and video analytics. In such a system IoT sensors (i.e., cameras) are connected to the broadband mobile network e.g., LTE. Captured video streams are sent to ME host where the application is running. The streams are processed by the application and in case an anomaly is detected, it will be sent to the core network. An overview of such deployment is illustrated in Fig. 3

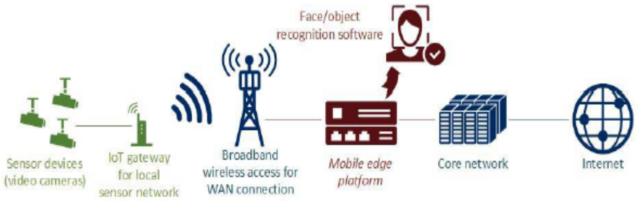
Face/object recognition software
Sensor devices (video cameras)
IoT gateway for local sensor network
Broadband wireless access for WAN connection
Mobile edge platform
Core network
Internet

Figure 2: Video Surveillance System in MEC [4]

D.2 IoT devices are Mobile

In this use case, IoT devices are connected to the broadband mobile network, and move across different cells. For a scenario such as mobile users, the aforementioned deployment triggers extra challenges in terms of backward compatibility. First, the user movement, calls for extra handover among edge servers. These servers are usually deployed at Base Stations (BS) or Access Points (AP). This mobility can cause complication due to diverse system configuration. Moreover, these movements pose interferences which result in deteriorating the communication performance. When communication performance is poor, the latency is negligible which will degrade user QoE.

Typically, mobile and session management is done at the core of the network e.g., by Serving GPRS Support Node (SGSN) in the current 3G/4G networks. Therefore, there is a need for non standard mechanisms in the legacy in order to support mobility.

Mobility management has been explored in cellular networks. Prior works [33, 34] modeled users' mobility by the connectivity probability or the link reliability according to information such as the users' moving speeds. However, these approaches cannot be directly applied to MEC.

Industry aims to bring core functions in the edge in order to support MEC in

IoT-based services. the current EPC solutions provides lightweight deployment of core network functions in proximity of the edge.

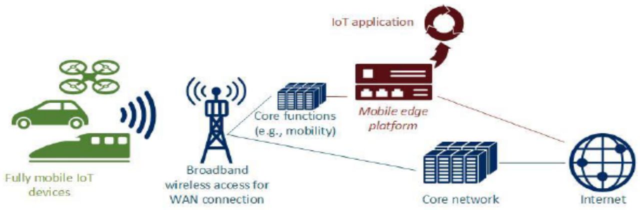

Figure 3: Mobile IoT use case in MEC [4]

yuyi et al. suggest new research directions in their survey considering mobility-aware server selection [6]. Here, we briefly summarize them.

1. **Mobility-Aware Online Prefetching:** While a user is mobile, the full information on his trajectory is not available. In the conventional design, the computation of the user's task will he handovered to the future server. However, this can bring additional computation offloading latency as well as stressing the MEC network. To overcome these obstacles, prior work [35] porpose a new solution referred as *online prefetching*. In this method, part of the computation task is prefetched to the potential servers at server computation time. However, this technique arises two important challenges. In order for prefetching to be efficient, user trajectory prediction must be precise. This precision depends on well-defined models and complex machine learning techniques. The secod challenge is selecting what data to be prefetched.

2. **Mobility-Aware Offloading Using D2D Communications:** D2D communications can be used in MEC network to help the mobility scenario. In this case, user's computation task is offloaded to his neighbors. Although D2D communications reduces energy consumption of transmission data, it arises new challenges to the design. First, the selection of neighbor nodes should be optimized according to user's trajectory information and computation capabilities. Second, massive D2D communication can introduce massive amount of interferences. Therefore, techniques such as interference cancellation can be applied.

3. **Mobility-Aware Fault-Tolerant MEC:** For real-time applications that are latency sensitive or computation demanding, any failure can cause non-negligible consequences. For instance, for a user who is using such an application in museum, any interruption can deteriorate user's QoE. Hence, three main areas are identified in the state-of-the-art: fault prevention, fault detection and fault recovery. To prevent failures in the system, extra offloading links can be deployed or cloud services can be used temporarily until the situation is stable again. To detect the failure, a periodically feedback can be collected from the nodes with respect to their channel quality. Later, for detected faults, recovery approaches should be performed. These

approaches can include offloading the task to the neighbor MEC systems via ad-hoc relay nodes [36].

4. **Mobility-Aware Server Scheduling:** Traditional server scheduling based on user priority can not be applied in mobile multi-user scenario in MEC. Therefore, the server needs to be adaptive and change its scheduling from time to time based on user's information. For instance, users with worst channel condition can have a higher offloading computation in order to complete their task on time. Another possible technique can be as follows: Predicting user's mobility and channel at the first place, and reserving resources based on this prediction. However, this approach has potential challenges in predicting and modeling the user mobility [37].

V Communication in MEC

Cloud Radio Access Network (C-RAN) and MEC are newly introduced to address limitations of Radio Access Network (RAN). The aim of C-RAN is to centralize BS function virtualization. Each of these technologies have a unique role in 5G networks. MEC servers are deployed in the BSs. This feature leads to low-latency in 5G networks. As discussed in previous sections, MEC uses RAN information such as user location, cell load, and allocated bandwidth to improve network by following services: (a) optimization of mobile resources (b) pre-processing and aggregation of large data before sending to cloud (c) context-aware services.

The concept of MEC and *fog computing* [38] are usually used interchangeably while they are slightly different. Fog computing is a general term that aims to bring storage and processing resources into the lower levels and usually owned by enterprise gateway devices. While MEC aims at improving the capabilities of RAN network at edge by introducing new interface between BSs and upper levels [39].

In the state-of-the-art MCC communication channel between devices and the server is abstracted as a bit pipe and modeled with a constant or random rate. While this models may be beneficial for large-scale cloud services, they cannot be used in MEC due to their simplification. One of the main goal in the MEC is to have an efficient air interface to support latency-sensitive tasks using a small-scale edge. Wireless communications are usually used in order to integrate control of computation offloading and radio resource management. For cases where the wireless channel is in deep fade, the computation offloading can be stopped or switched to an alternative medium. Furthermore, there should be a trade-off between the wireless transmission and computation offloading, since higher transmission offers higher data rate, but causes larger energy consumption. Thus, the design needs to be adaptive to the varying channel based on the precise channel state information. [6].

Servers in MEC are small-scale data centers and deployed by cloud computing. These servers can be located with BSs, APs such as Wi-Fi routers. In MEC, typically the communication is between the APs and devices. In addition, D2D communication between devices can be supported. This will provide peer-to-peer communication among neighbor nodes, and can also be used for load balancing. The role of the AP is to provide wireless interface for the MEC and access to the remote data center using backhaul links.

VI conclusion

Edge computing brings datata and services closer to the end users to ensure lower latency for data-intensive applications, lower the required bandwidth and ensure privacy. In this paper, we studied the edge computing architecture, use-cases, and applications. We classified the state-of-the-art in edge computing according to the application domain. The application domain areas include services such as smart cities, video analytics, real-time applications, privacy, resource management. Furthermore, in each section we identified and expanded on open research challenges and current directions.

References

[1] Ericson. Ericcson mobility report. 2013.

[2] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4), 2009.

[3] Milan Patel, B Naughton, C Chan, N Sprecher, S Abeta, A Neal, et al. Mobile-edge computing introductory technical white paper. *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.

[4] Dario Sabella, Alessandro Vaillant, Pekka Kuure, Uwe Rauschenbach, and Fabio Giust. Mobile-edge computing architecture: The role of mec in the internet of things. *IEEE Consumer Electronics Magazine*, 5(4):84–91, 2016.

[5] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5g. *ETSI White Paper*, 11(11):1–16, 2015.

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

[7] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

[8] Arif Ahmed and Ejaz Ahmed. A survey on mobile edge computing. In *Intelligent Systems and Control (ISCO), 2016 10th International Conference on*, pages 1–8. IEEE, 2016.

[9] Geoff Simmons, Gillian A Armstrong, and Mark G Durkin. An exploration of small business website optimization: enablers, influencers and an assessment approach. *International Small Business Journal*, 29(5):534–561, 2011.

[10] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 2019.

[11] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *computer*, 50(10):58–67, 2017.

[12] Qingyang Zhang, Hui Sun, Xiaopei Wu, and Hong Zhong. Edge video analytics for public safety: A review. *Proceedings of the IEEE*, 107(8):1675–1696, 2019.

[13] Lur Tze Hsien and Indriyati Atmosukarto. Video analytics in train cabin using deep learning. In *2019 4th International Conference on Intelligent Transportation Engineering (ICITE)*, pages 94–98. IEEE, 2019.

[14] Humans can't watch all the surveillance cameras out there, so computers are, 2019.

[15] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *computer*, 50(10):58–67, 2017.

[16] British transport police: Cctv, 2019.

[17] Roelof Kemp, Nicholas Palmer, Thilo Kielmann, Frank Seinstra, Niels Drost, Jason Maassen, and Henri Bal. eyedentify: Multimedia cyber foraging from a smartphone. In *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on*, pages 392–399. IEEE, 2009.

[18] Shanhe Yi, Zijiang Hao, Qingyang Zhang, Quan Zhang, Weisong Shi, and Qun Li. Lavea: Latency-aware video analytics on edge computing platform. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–13, 2017.

[19] Hannaneh Barahouei Pasandi and Tamer Nadeem. Collaborative intelligent cross-camera video analytics at edge: Opportunities and challenges. In *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pages 15–18, 2019.

[20] Hannaneh Barahouei Pasandi and Tamer Nadeem. Convince: Collaborative cross-camera video analytics at the edge. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2020.

[21] Hannaneh Barahouei Pasandi and Tamer Nadeem. Convince: Collaborative cross-camera video analytics at the edge. *arXiv preprint arXiv:2002.03797*, 2020.

[22] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[23] Ejaz Ahmed and Mubashir Husain Rehmani. Mobile edge computing: opportunities, solutions, and challenges, 2017.

[24] Somayya Madakam and R Ramaswamy. The state of art: Smart cities in india: A literature review report. *International Journal of Innovative Research and Development*, 2(12), 2013.

[25] Michael Till Beck, Martin Werner, Sebastian Feld, and Thomas Schimper. Mobile edge computing: A taxonomy. 2014.

[26] MECISG ETSI. Mobile edge computing (mec); framework and reference architecture. *ETSI, DGS MEC*, 3, 2016.

[27] Le Liang, Hao Ye, and Geoffrey Ye Li. Toward intelligent vehicular networks: A machine learning framework. *IEEE Internet of Things Journal*, 6(1):124–135, 2018.

[28] Hannaneh Barahouei Pasandi and Tamer Nadeem. Challenges and limitations in automating the design of mac protocols using machine-learning. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 107–112. IEEE, 2019.

[29] Hannaneh Barahouei Pasandi and Tamer Nadeem. Poster: Towards self-managing and self-adaptive framework for automating mac protocol design in wireless networks. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 171–171. ACM, 2019.

[30] Muhammad Zohaib Anwar, Zeeshan Kaleem, and Abbas Jamalipour. Machine learning inspired sound-based amateur drone detection for public safety applications. *IEEE Transactions on Vehicular Technology*, 68(3):2526–2534, 2019.

[31] Hannaneh Barahouei Pasandi and Tamer Nadeem. Mac protocol design optimization using deep learning. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 709–715. IEEE, 2020.

[32] Ke Zhang, Yuming Mao, Supeng Leng, Sabita Maharjan, and Yan Zhang. Optimal delay constrained offloading for vehicular edge computing networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[33] David Lopez-Perez, Ismail Guvenc, and Xiaoli Chu. Mobility management challenges in 3gpp heterogeneous networks. *IEEE Communications Magazine*, 50(12), 2012.

[34] Aleksandar Damnjanovic, Juan Montojo, Yongbin Wei, Tingfang Ji, Tao Luo, Madhavan Vajapeyam, Taesang Yoo, Osok Song, and Durga Malladi. A survey on 3gpp heterogeneous networks. *IEEE Wireless Communications*, 18(3), 2011.

[35] Seung-Woo Ko, Kaibin Huang, Seong-Lyun Kim, and Hyukjin Chae. Live prefetching for mobile computation offloading. *IEEE Transactions on Wireless Communications*, 16(5):3057–3071, 2017.

[36] Dimas Satria, Daihee Park, and Minho Jo. Recovery for overloaded mobile edge computing. *Future Generation Computer Systems*, 70:138–147, 2017.

[37] Yuan Zhang, Jinyao Yan, and Xiaoming Fu. Reservation-based resource scheduling and code partition in mobile cloud computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 962–967. IEEE, 2016.

[38] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[39] Tuyen X Tran, Abolfazl Hajisami, Parul Pandey, and Dario Pompili. Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, 55(4):54–61, 2017.