Journal of Information Processing Vol.26 87–97 (Feb. 2018)

[DOI: 10.2197/ipsjjip.26.87]

Invited Paper

Multi-access Edge Computing: A Survey

Hiroyuki Tanaka[1,a)] Masahiro Yoshida[1] Koya Mori[1] Noriyuki Takahashi[1]

Received: July 4, 2017, Accepted: November 20, 2017

Abstract: Multi-access Edge Computing (MEC) can be defined as a model for enabling business oriented, cloud computing platform within multiple types of the access network (e.g., LTE, 5G, WiFi, FTTH, etc.) at the close proximity of subscribers to serve delay sensitive, context aware applications. To pull out the most of the potential, MEC has to be designed as infrastructure, to support many kind of IoT applications and their eco system, in addition to sufficiently management mechanism. In this context, various research and standardization efforts are ongoing. This paper provides a comprehensive survey of the state-of-the-art research efforts on MEC domain, with focus on the architectural proposals as infrastracture, the issue of the partitioning of processing among the user devices, edge servers, and a cloud, and the issue of the resource management.

Keywords: edge computing, MEC, distributed processing, multi-tier cloud computing, Internet of Things (IoT)

1. Introduction

In the past decade, cloud computing has been booming. The design principle of the cloud is concentration. More computation and storage are gathered to the data centers, and application users use them on demand through ubiquitous networking, thanks to the commoditization of wireless access. This ecosystem has been greatly successful because of economies of scale.

However, the emergence of the Internet of Things (IoT) brings new requirements and challenges to cloud computing. Firstly, several promising IoT applications need very short delay time. Accroding to a ITU-T Technology Watch Report [1], a typical control interval of industrial robot in a closed-loop system is roughly a millisecond, and a human visual reaction time is in the range of 10 milliseconds, for example. For the application that requires these short delay, packets round trip to the cloud is very likely too long. Secondly, heavy traffic volume produced by enormous number of IoT terminals will put too much burden on the network infrastructure.

To cope with those issues, the concept of the edge computing is emerging. The term "edge computing" is not new; the first appearance in the literature [2] seems to date back to 2001. In 2002, Microsoft published technical report titled "Enabling rich content service on the edge [3]," and Akamai announced "EdgeComputing [4]" in 2003. But they were published before cloud and IoT, and were different from the recent proposals. In 2009, a visionary and pioneering article by Prof. Satyanarayanan of CMU, titled "The case for vm-based cloudlets in mobile computing [5]," was published and placed basis for today's edge computing concept. Since then, many research have been conducted actively, such as Refs. [6], [7] and [8], among others, and the paradigm gained momentum. While at the same time, it becomes apparent that, to pull out the most of edge computing potential, it has to be infrastructure that is widely distributed, works together with network equipment in reasonable manner, is shared among multiple users, and is sufficiently managed.

In response to this, in 2014, ETSI, European Telecommunications Standards Institute, organized ISG (Industry Specification Group) for Mobile Edge Computing (MEC) [9] for global standardization. As the name suggests, MEC ISG first targeted edge computing combined with cellular wireless (LTE and 5G). Later, in 2016, the ISG changed its name to Multi-access Edge Computing (abbreviation remains same "MEC") to extend its scope to cover other access technologies like WiFi and fixed. So far, the ISG published a number of documents such as white papers [10], [11], service scenarios [12], requirements [13], and a reference architecture [14], among others. Other collaboration efforts have been initiated, as mentioned later in this survey. While there are numerous proposals related to edge computing, henceforth, we use MEC as a synonym for cloudlet, fog computing, and other related architectures, and regard the words interchangeable.

This survey is organized as follows. First, definition of edge computing and similar architectures are presented, along with its foreseen advantages. Next, several service scenarios are introduced and related application proposals are surveyed. It is followed by description of ETSI MEC's reference architecture that is a major standardization effort of edge computing. Then, we introduces several new and cutting-edge research works in the areas that we believe important to deploy MEC as a major part of future infrastructure, namely, the architectural proposals for edge computing, the issue of the partitioning of processing among the user device, edge servers, and a cloud, and the issue of the resource management. In this survey, we do certainly not intend for completeness, but we refer some of good surveys in the conclusion.

2. Definition of Edge Computing

To suffice for enormous computation requirement, cloud has

[1] NTT Network Innovation Laboratories, Nippon Telegraph and Telephone Corporation, Musashino, Tokyo 180–8585, Japan
[a)] tanaka.hiroyuki@lab.ntt.co.jp

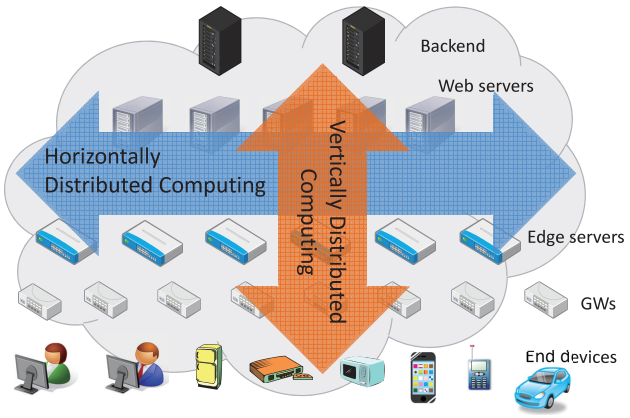

Fig. 1 Vertically and horizontally distribited computing.

evolved in two dimensions; horizontally and vertically. **Figure 1** represents a general idea of the dimensions. Horizontally distributed computing in cloud is, in short, to replicate many servers that run mostly same program, for the purpose of scaling out and redundancy. Vertically distributed computing, also commonly known as multi-tier cloud computing [15], is to decompose computation to a series of sub-processes and connect them to form a processing pipeline. This is good for segregation and structuring of application logic. Horizontal distribution and vertical distribution can be used in combination. Most succeeded vertical distribution is web three-tier model [16] which consists of web servers, application servers, and database servers. Conventionally, those tiers are placed in one place; for example in the same datacenter. However, some functionality might be placed other than the datacenter, namely, on the midway servers in between the datacenter and clients, and the whole infrastructure constitutes a multi-tier structure of computation-capable nodes and communication links. The midway servers are supposed to be located in proximity of users or end devices, near the edge of networks. This architecture concept is called edge computing, and the midway servers are often called edge servers, though several other names are also used, such as "cloudlet [5]" and "fog computing [17]."

Satyanarayanan et al. [5] define cloudlet as follows: *"a mobile user exploits virtual machine (VM) technology to rapidly instantiate customized service software on a nearby cloudlet and then uses that service over a wireless LAN; the mobile device typically functions as a thin client with respect to the service."* This is a kind of archetype of today's edge computing concept. Although it states *"over a wireless LAN,"* cloudlet is now extended to utilize with cellular networks [18].

The other well-known proposal is Cisco's fog computing. Original proposal [17] was in 2012. It states *"Fog Computing is a highly virtualized platform to provide compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network."* Beside computation, fog also intends to help inter-device networking. Fog computing and edge computing are regarded as interchangeable each other these days.

In MAUI [6], remote execution technique is used to reduce the energy consumption of mobile devices, through fine-grained code offload per method. In MAUI, it is decided at runtime whether a method is executed locally (on the device) or remotely (at an edge server), to maximize energy savings while satisfying a latency limitation, under the mobile device's current connectivity constraints. To ease the programmer's burden, it uses code portability; namely, two versions of a smartphone application, one for execution on the smartphone and the other for edge servers. For the similar purpose, CloneCloud [7] uses an application-level virtual machine that runs on edge servers. In CloneCloud, offloading granularity is per thread. In Virtual Smartphone [8], most of the application processing is offloaded to an edge server The user device (smartphone) application receives the screen output of a virtual smartphone running at the nearby edge, in similar way as conventional thin-client technology. These early proposals, except fog computing, focus on user equipment end devices like smartphones and tablets. But, the effect of offloading by edge servers remains as same with IoT devices.

ETSI ISG MEC defined that, in its first introductory whitepaper [10], *"Mobile-Edge Computing provides IT service environment and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile device."* Later, the definition is slightly broadened in Ref. [11], *"Edge Computing refers to a broad set of techniques designed to move computing and storage out of the remote cloud (public or private) and closer to the source of data,"* to accomodate various access technologies.

The main purpose of the edge computing is summarized as follows (specific service scenarios are described later in Section 3);

- Real-time distributed computing: edge servers can reply request from end devices with a shorter response time than central servers.
- Reduction of processor load in end devices (offloading): edge servers can assist heavy interactive application process, which suppress processing load on performance-poor end devices.
- Localization of data: in cases such as M2M applications and analysis of data from geographically distributed sensors, localization of data processing on a near by edge server can optimize network traffic and cloud server's load.

For the above purpose, the advantages of edge computing fall roughly into three categories, low latency for application processing, client offload, and efficient utilization of infrastructure resources. It means there are multiple metrics for the performance and effectiveness of edge computing.

(1) From the view point of end users/devices, delay and battery power usage are important.
(2) From the view point of service providers such as telecommunication carriers and Internet Service Providers (ISPs), it is important to avoid the congestion or exhaustion of servers and networks resource, and reduce the cost of servers and networks, while maintaining users' satisfaction.
(3) In addition, from the view point of service developers, it is necessary to have a good eco system that invites good applications from third party developers.

Similarly, Shi et al. [19] list the optimization metrics as follows: (a) Latency, caused by both computation and networking (especially WAN delay), (b) Bandwidth, (c) Energy consumption on

user devices, which is tradeoff between the computation energy consumption and transmission energy consumption, and lastly, (d) Cost to build and maintain, while the improved user experience can result in higher revenue.

There are tradeoffs among the performance metrics. For example, Takahashi et al. [20] describe the case when a part of client processing is offloaded to an edge server, where the original client program is split into two parts, and one of the parts is moved to an edge server. Calculation delay on the client decreases, while new calculation delay arises on the edge server. Since the server is faster, the reduction in delay is a linear to the amount of offloaded computation. As for the delay related to traffic to/from the client, the newly increased traffic between the client and the edge server travels over a wireless link. So the delay in communication will increase. But, the increase is likely less than proportional to the increase of the traffic volume since there is a fixed overhead that is independent to the volume, such as session establishment.

3. Service Scenarios and Applications

One of the ETSI ISG MEC's outcome, "Service Scenarios [12]," categorizes candidate services into seven scenarios. This section summarizes them and refers some of new academic proposals that fall into each category.

3.1 Intelligent Video Acceleration

This service scenario is to optimize the quality of video content delivery to mobile devices by guiding video sources (like YouTube) as to wireless access environment, thus to improve delivery efficiencies. In this service scenario, a Radio Analytics application is supposed to be located at RAN (Radio Access Network) and provide the video server with information of the estimated available throughput at the radio downlink. The information can be utilized for adjusting TCP congestion control mechanism or for ensuring application level coding parameters. In this case, an edge server for the analytics is supposed to get radio status from cellular nodes like base station.

Wang et al. [21] realize a prototype in this scenario in straightforward way. The system is implemented with 4G LTE emulator, and the MEC server function is added to a component of the eNodeB (Evolved Node B which is Base Station component in the LTE cellular mobile phone networks). The MEC server examines the Channel Quality Indicator (CQI) that is measured and sent back by an end user equipment in response to reference signals from eNodeB. Based on CQI, channel information is sent to the video server, and the video system bitrate is dynamically changed using MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH). Their experimental results is good; in short, the proposed method give low latency for various against various fading profiles, while producing a reasonable throughput which is closer to the upper bound of high constant bitrate streaming case.

In the Superfluidity project [22], the design of a Video Streaming service utilizing MEC is studied [23]. In streaming service, several different application container protocols such as Apple HTTP Live Streaming, MPEG-DASH, Adobe HTTP Dynamic Streaming, etc. are used. Upon request from the client, an original video content is split into small segments and each segment is formatted for the requested protocol. This operation is called trans-multiplexing (transmuxing). Conventionally, the transmuxing is performed at an origin server, and CDN caches formatted container. In the work, the transmuxing is offloaded to edge; the origin server send unformatted container-independent segment to an edge server, and the edge server, while caching the segment, performs transmuxing. Experimental shows that this approach results in lower latency and gives a better throughput compared to the traditional CDN, for video that is streamed in a different format before. They also make the offloading dynamic using the Reusable Functional Blocks (RFB) and Docker [24] container mechanism.

Tran et al. [25] investigate MEC usability for a joint collaborative caching and processing problem (JCCP) for on-demand video streaming. Here, MEC computational capability is used for transcoding of a video to different bitrate version to satisfy user requests, so that users can receive videos that are suited for their network condition and capabilities of the user terminals. The word *collaborative* means that multiple MEC servers, connected by backhaul network, assist each other for content caching and transcoding. The problem is formulated as an Integer Linear Program that minimizes the backhaul network cost, subject to the edge servers' cache storage and processing capacity constraints. Due to NP-completeness of the problem, they propose heuristics that works very well. The performance of the proposed method is close to offline-calculated optimal, in terms of cache hit ratio, average access delay, and backhaul traffic load.

3.2 Video Stream Analysis

This service scenario is for applications like a video based monitoring or surveillance system. For example, suppose vehicle license plate recognition that monitors vehicles entering and exiting an area, for security purposes, and so on. A MEC server near the camera can be used to capture and analyze the video, then send the recognized plate number to a cloud. The size of data to the cloud is far smaller in size compared to the original video. The mechanism for video analysis itself remains similar to the cloud based video analysis [26]. Edge based Analytics is comprehensively described in Ref. [27]. Several vendors are working on this service scenario, too [28], [29].

3.3 Augmented Reality

For this service scenario, suppose that a visitor to a museum, for example, holds their smartphone towards a particular point of interest with the museum application. The smartphone's camera captures the point of interest and the application displays additional information regarding what the visitor is viewing, using augmented reality (AR) system. The use of a MEC server is advantageous as the AR information is localized. In addition, AR generation should follow the change of user's look, so that low latency and responsiveness is required.

Verbelen et al. [30] implement an AR application featuring markerless tracking and object recognition. The application tracks feature points in the video frames and overlay 3D objects on the screen. They split the application logic into several components, and dispatch each components based on the real time

requirement, whether to nearby edge server or central cloud. As the result, VideoSource and the Renderer are fixed on the mobile device, Object Recognizer and Mapper are in a cloud, and remaining Object Tracker and AR Relocalizer are executed in the edge server.

Chen et al. [31] propose a cognitive assistance application using Google Glass with cloudlet. In their implementation, named "Gabriel," the glass is used as input device of viewing (video), position (GPS), and acceleration information of the user. These data are sent to the cloudlet server, and several subsystems such as face recognition, object recognition, motion classifier, and augmented reality, among others, are executed on the server. Each subsystem is realized as a separate virtual machine on enhanced OpenStack platform.

Dolezal et al. [32] use an Augmented Reality application for performance evaluation of computation offloading from mobile device to an edge server. This work is described later, in Section 5.2. Beside AR, Mangiante et al. [33] propose MEC use for an interesting kind of Virtual Reality application. An edge server is used to perform Field Of View rendering from high-definition (4k class) 360° live stream. Though preliminary, results show the immediate benefits in bandwidth saving while maintaining higher frame rate.

3.4 Assistance for Intensive Computation

This is to offload intensive computation from the end devices to maximize battery life or to simplify devices (especially, low cost sensing devices). The image recognition, already appeared in the AR service scenario, is example of such computation. Wearable devices for gaming, environmental sensors, and security applications could be other examples where offloading can be useful.

Takahashi et al. [20] propose Edge Accelerated Web Browsing to accelerate the web applications execution by offloading a tiny device such as HDMI stick. Appplications that are mentioned in other categories have the aspect of intensive computation, so we omit here to avoid duplication.

3.5 Enterprise Deployment of MEC

In this service scenario, MEC is expected to realize a transparent breakout from within the mobile carrier's RAN to the enterprise network. This is because, with the success of cloud computing, many enterprise services are migrating to cloud based platforms, and users are willing to connect their own devices to the enterprise network as well as public network, while maintaining security and performance requirements. One simple example is integration of an IP-PBX with a MEC platform that could provide seamless service between a telecom operator small cell and the enterprise WLAN network. While local breakout functionality has been in standardization by 3GPP [34], the MEC might be add capabilities to perform enterprise-specific services and policies, such as access control and service differentiation, etc.

Another application is to put an edge server along the production lines in a manufacturing factory. The edge server is used to collect information from Computer Numerical Control devices, manufacturing robots, peripheral devices, and the like. The edge server performs advanced analytics and real time feedback control when necessary. This smart factory scenario is studied in Refs. [35] and [36], for example, and there are efforts for initial deployment in actual factories, such as Refs. [37], [38] and [39].

3.6 Connected Vehicles

Cars and other vehicles are expected to become more "connected" in the next decade. To connect, technologies such as Dedicated Short Range Communications are utilized for short distance connectivity, and cellular networks, namely LTE and 5G, are for long distance and wider coverage. Communication of vehicles to vehicles (V2V) and vehicles to roadside-sensors (V2I) is intended for increased safety, efficiency, and convenience of automotive. Information about nearby road hazards, vehicles next behavior, congested road, or even unoccupied parking location, etc. will be exchanged. These information are usually locally usable. In other words, they are valuable within some limited proximity, so that the processing by a nearby MEC server has significant advantage. In addition, MEC can be used to provide the hosting services for the application that requires low latency.

Regarding the connected vehicles, a lot of new services and business opportunities is expected, thus the service scenario is attracting a great deal of attention. There are comprehensive survey works [40], [41] which are specific to the MEC related research in this area, so we do not mention individual papers here. The industries also pay attention for MEC application to the connected vehicles and activities to foster the ecosystem is starting in some consortiums [42], [43], [44], etc.

In research that relates to vehicles, actual moving behavior of vehicles is helpful for realistic evaluation. For this purpose, SUMO: Simulation of Urban MObility [45], [46] data is widely used.

3.7 IoT Gateway

With the growth of the Internet of Things, many and various devices become connected, and they exchange messages among them and with a cloud. This is a vast service category and many new and useful use cases are anticipated. Basically, a MEC server acts as a gateway to aggregate the messages from IoT devices nearby. In addition, the MEC server can pre-process each message and send only meaningful messages (for example, when a sensor value changes more than specified threshold). This will significantly reduce communication and processing overhead in a cloud. Moreover, depending on the nature of some of the devices that are connected, a real time processing capability, or functionality for the proximity-based device group formation, etc. are needed for efficient service, where MEC is expected to be useful.

Osmotic computing [47] proposes a paradigm for the efficient execution of IoT services and application at the network edge. Its design concept is based on three tiers for application processing, namely, IoT devices, edge servers, and public/private cloud, as similar to other proposals. Applications are decomposed into microservices that are tailored and deployed dynamically either at edge or in cloud. Like *osmosis* in the context of chemistry, the dynamic management of resources in cloud and edge is performed to achieve the balanced deployment of microservices while satisfying resource constraints and application needs. This balance

is dynamically tunable depending on configuration for the resource involvement, so that the operator can determine whether microservices should migrate from cloud to edge or vice versa. Osotic computing is designed in application-agnostic approach, utilizing lightweight container-based virtualization technologies such as Docker [24] and Kubernetes [48] for the deployment of microservices.

Sapienza et al. [49] examine a scenario that exploits the MEC for detecting abnormal or critical events such as terrorist threats, natural and human-caused disasters. In the paper, three sources of information are assumed; personal devices like smartphones, video surveillance system deployed in the city, and wireless air quality sensor system. In their scenario, MEC servers performs two services; Mash-Up Service that monitors and analyze the data from information sources, and Alert Notification Manager that make notification messages and send them to neighbor Base Transceiver Stations or eNodeBs.

EdgeIoT [50] explores user privacy issue in mobile edge computing for the internet of things. In the proposal, each user's IoT devices are associated with a proxy VM (located in a fog/edge server). The proxy VM collects, classifies, and analyzes the devices' raw data streams, converts them into metadata, and transmits the metadata to the corresponding application VMs (which are owned by IoT service providers and used by users in shared manner). The metadata, which is exchanged among proxy VMs and application VMs, is generated from the raw data streams so at to not violating user privacy. For instance, in the terrorist detection application, only the locations and timestamps of the matched photos/videos are uploaded to the application VM.

4. MEC Standardization

In pioneering work that we mentioned [5], [6], [7], [8], [17] in the previous section, each of them has proposed its own MEC model or architecture with slightly different motivation and scope. However, with various research and standardization activities that followed, it becomes evident that the MEC architecture needs to be designed as infrastructure, to support many kind of IoT applications and their eco system.

The standardization effort in ETSI ISG MEC has started in 2014. **Figures 2** and **3**, quoted from Ref. [14], are the MEC framework and the MEC reference architecture of current, respectively. The Standardization in ETSI ISG MEC is still work in progress, and it should be noted that the reason we described ETSI MEC architecture here is not because it is definitive one. Rather, we describe it here to show the broad extent of standardization consideration for the purpose of defining MEC as worldwide usable IoT infrastructure. The design is influenced by the preceding establishments in areas of the Software Defined Networks and the Network Function Virtualization, that are also standardized in ETSI [51] and being adopted in telco's commercial networks gradually.

The MEC framework, in Fig. 2 shows the abstract functional entities in the MEC architecture. The entities work in either the system level, Multi-access Edge (ME) host level, or network level. The ME host level, middle one, is split into the ME platform, the ME applications, and the virtualization infrastructure. It

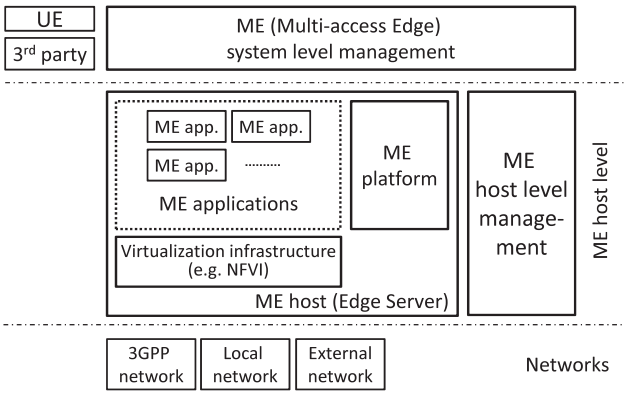

Fig. 2 The ETSI MEC framework (based on Ref. [14]).

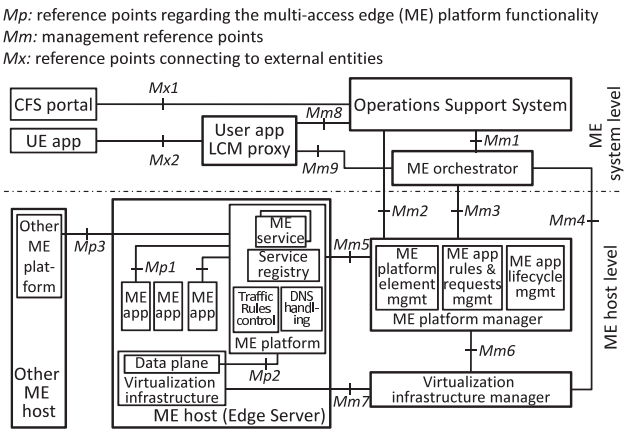

Fig. 3 The ETSI MEC reference architecture (based on Ref. [14]).

is supposed to use NFV infrastructure (NFVI) in ME host. For the packaging, deploying and execution of ME applications, the VM virtualization is used. The networks levels repesents the MEC connectivity of the access networks; namely the 3rd Generation Partnership Project [34] (3GPP) cellular network, the local networks (enterprise LAN, etc.), and the external network (the Internet), by means of WiFi and fixed line (e.g., FTTH). The ME system level on the top in the figure provides the view of whole ME system to UEs and external 3rd party (applications in cloud, for example). It consists of the ME host and the ME management system. The latter is necessary to execute ME applications within an operator network.

In Fig. 3, the functional entities are defined in more detail with the relations among them and respective reference points. The ME host provides computing, storage, and network resources for the ME application, through VM virtualization. OpenStack is considered as the current best practice, though lightweight Container approach is also under study. The ME platform in the MEC host represents a collection of essential functionalities to run applications on a ME host and to enable ME applications to discover and consume the ME services. The platform also includes the function for the traffic forwarding that is necessary to steer the traffic among the applications, services, and networks. The right half of the figure depicts the management and the operation functionalities and several reference points for the purpose are defined, which are necessary for coordinated operation between operators and service propviders. For whom being interested in, it is recommended to refer to the original document in Ref. [9], as

it is on-going effort.

3GPP, the major international standardization organization for LTE and 5G etc., is also paying attention to MEC, and the related discussion and the liaizon with ETSI ISG MEC are already active. In 3GPP specification series 23 (TS23), the following four items that are relevant to MEC are examined. Namely, (1) Local breakout at User Plane Function selection, to steer the traffic to the relevant ME host according to the packet header and contents, (2) QoS and policy controll for MEC traffic at Session Management Function, (3) utilization of function discoverymechanism known as Network Repository Function, to find appropriate ME host, and (4) use of NW function API for lower latency and wider bandwidth, provided by Network Exposure Function.

In addition to ETSI and 3GPP, collaborative efforts are also activated in the form of consortium, namely, Open Fog Consortium [52] by Cisco et al., Open Edge Consortium [53] (OEC) by Carnegie Mellon University et al., Edge Computing working group in Telecom Infra Project [54] (TIP) by Facebook et al., and Edge Computing Consortium [55] (ECC) by Huawei et al., among others.

5. Recent Researches for MEC Infrastructure

5.1 Architectural Studies

As stated in the previous section, MEC does not intend a standalone solution for a single specific application, rather it aims to be infrastructure to provide computation and storage capability in close proximity of users and devices for broad range of applications. In this regard, deliberation in MEC architectural design is of most importance. There are many proposals and studies, either based on ETSI MEC/NFV or not. This section introduces a few novel and interesting works among them.

Tran et al. [56] present architectural view of MEC in future 5G cellular networks. 5G aims ultra-low latency of 1 millisecond end-to-end round trip [57], so that it is expected to be one of most suitable access networking technologies for MEC in realizing real time application processing. In the study, MEC servers are implemented directly at 5G base stations (BSs) to fulfill the stringent low-latency requirement. They compare MEC with Cloud RAN (C-RAN. RAN stands for radio access network), that has been emerging paradigm in cellulara architecture. The two paradigm move computing capabilities in a different direction; C-RAN aims to centralize base station functions via virtualization and SDN, while MEC is, in a sense, distribution of functions toward network edge. They summarize that MEC and C-RAN do not contradict but rather are complementary each other. For example, an application that requires very low delay can have one or few time-critical components running in MEC and other components in the cloud. They introduce three case studies, mobile edge orchestration, collaborative video caching and processing, and two-layer interference cancellation, in MEC enabled 5G systems.

The EU funded SESAME project [58] proposes the Cloud-Enabled Small Cell (CESC) concept and investigates the placement of network intelligence and application in the network edge. One interesting point is that the SESAME is designed to enable CESC as multi-operator (multi-tenancy) entity. That means multiple network operator will be able to use the SESAME platform, each one using its own "slice" of network including MEC capability. The project presents a white paper [59] that contains the reference model and the architecture design, that are mostly aligned with ETSI framework, in addition to its feature description and implementation of the prototype.

Network Function Virtualization (NFV) is considered to provide a technological basement to realize MEC infrastructure. Cziva et al. [60] propose Glasgow Network Functions (GNF) for the NFV platform for MEC. GNF is container-based lightweight module encapsulation and provides fast instantiation time with low resource overhead. It supports VNF roaming to follow users between cells seamlessly. They present realistic spec of possible edge server HW specifications, compare GNF with existing VNF approaches in regard to MEC, and give a few use cases. In similar context, Carella et al. [61] study application of NFV in MEC, but it focuses the Management and Orchestration (MANO) services for MEC service modules. Based on the Open Baton [62] framework for NFV MANO, it achieves cross-domain orchestration that supports NFV and MEC, by making use of the existing functional elements in MANO, namely the Virtual Network Function Manager (VNFM) and the Virtualized Infrastructure Manager (VIM).

In LTE and 5G networks, there are optical backhaul that connects base stations to operator's core network. In the backhaul, use of Ethernet based technologies, e.g., Ethernet passive optical network (EPON) and 10G-EPON, is regarded as a compelling solution and widely used recently. In this fiber-wireless (FiWi) access networks, access points or BSs are collocated with optical network units (ONUs). Rimal et al. [63] tackle the important issue that is the integration of MEC into this existing FiWi infrastructure. In the article, a few architectures of MEC over FiWi are presented with the candidate place to put MEC servers. Then, it proposes a TDMA-base unified resource management scheme that allows coexistence of conventional non-MEC traffic and MEC-related traffic. Because of the schedule nature of TDMA, the scheme allows MEC-assisted user devices go into sleep mode during Other devices' timeslot. The performance evaluation, though analytical, shows that the proposed method achieves a good response time efficiency and reasonable MEC packet delay, as well as prolonged battery life of MEC-assisted devices.

Though it is not architectural, Orsini et al. [64] raise an important issue when MEC services are not yet fully deployed but available in limited places. In the scenario, a user has a handheld device with cellular and WiFi connectability. Through the cellular service, an application running on the device is offloaded to cloud over internet, while on the other hand, it is offloaded to a nearby MEC server via WiFi connection when in WiFi hotspots. The cloud server and the MEC servers provide a same set of application services, that are different image filters in this case. To enable switching automatically between cellular and WiFi, they implement "CloudAware" that is a middleware running on Android OS. The CloudAware has context adaptation feature to select appropriate connection selection and offload timing, by means of its history based prediction of user future movement, available

bandwidth, and the execution time of an task. For realistic evaluation, they use Nokia Mobile Data Challenge (MDC) dataset [65] that contains activity records of data from smartphones of almost 200 volunteers over 18 months in Lausanne area, such as smartphone location, Cellular/WiFi connection status, device battery level, application usage, and so on. Evaluation is done in simulation in which each application tasks are judged either success or failure; the task may fail because of various reason such that the device battery runs out, the nearby MEC server is overloaded, or connection is lost by user's movement. Although the simple setup, the simulation results shows that the CloudAware provide speedup of more than double, while maintaining a similar success rate, compared with the device local execution.

5.2 Application Partitioning and Performance Studies

There are several optimization metrics in MEC systems, namely latency, bandwidth consumption, battery usage in user devices, and infrastructure cost, among others. The tradeoff among them changes depending on not only MEC implementation but also workload partitioning among a user device, an edge server, and a cloud. Investigation and understanding the tradeoff mechanism is very important for MEC application and actual deployment. The issues in workload partitioning are described and studied in detail in Ref. [66], chapter 2. It sets the following question for the problem, quoted; *Given a specific application state and a specific computational environment, which portions of the application should run on the mobile computer and which should run on remote infrastructure?* There are several works that tackle the question.

Dolezal et al. [32] present evaluation of computation offloading from mobile device to an edge server. In the experiment, they use Android smartphones, Intel Xeon workstation for the edge server, and WiFi network to connect them. The partition of the application into modules seems to be done beforehand by the programmer, and computation-intensive modules are marked using Java annotation. Their offloading framework, called UE stack, monitors and intercepts attempts to execute the application module, and it is decided whether the offloading should be performed instead of UE local execution. For the evaluation, they use an AR application that discovers places of interest visible in the view of the device's camera and show additional text annotation information as an overlay. The application is computation-intensive and the workload depends on the scene; the workload increase with the *Discovery Range* that is the maximum distance from the device where the places are the subject of discovery. The revaluation result shows that the offloading decreases latency drastically compared to the UE local execution, and the effect is more distinctive when the workload is high, namely, when the Discovery Range is large. The latency heavily depends on the network throughput. The down side is increased traffic for offloading, though the increase is rather modest. For this application, the offloading also reduces the energy consumption on the smartphones, despite additional requirement for offload communication.

Hu et al. [67] present the comparison of three cases that are UE local execution (no offload), offload to a cloud (Amazon EC2), and offload to a MEC server (cloudlet running on 2.7 GHz quad-core workstation). Three applications are evaluated in the experiments; a face recognition, augmented reality, and physics-based computer graphics. They also use heavy benchmarking programs such as Linpack (numerical linear algebra) and PI (calculating π up to 2 million digits of precision). The applications are partitioned by COMET [68]. LTE and WiFi are used for network accesses that give different round trip latency, ranged from a few milliseconds for the cloudlet via WiFi, to more than 100 milliseconds for the cloud over LTE access. The results give in the paper is very informative. In comparison with the cloud that has superior performance (almost double), the edge computing is able to improve response time and energy consumption significantly, both with WiFi and LTE.

Wang et al. [69] investigate the partial computation offloading problem that is to optimize the offloading ratio with two minimizing objectives: energy consumption of user devices and latency of application execution. Moreover, dynamic voltage scaling technology is incorporated in the study to fully utilize advanced chip functionality for low energy consumption in mobile devices. The both minimization problems as formulated as nonconvex problems at first, then transformed to a convex problem. This is an analytical method and gives an optimal offloading ratio only; partitioning mechanism is out of the scope. Mao et al. [70] discuss power-delay tradeoff in analytical way, too, for multi-user mobile-edge computing systems. In summary, the average execution delay increases as the power consumption in the user devices decreases when more workload is offloaded to the MEC system. Mao et al. [71] investigate an interesting case in which the end devices possess energy harvesting (EH) capability. It discusses an computation offloading strategy for EH devices, and proposes a low-complexity online algorithm.

In the software engineering community, the microservice-oriented software architecture has drawn an attention, where an application is constructed as a collection of loosely coupled, relatively small software modules called *microservices*. It improves modularity of software components and makes the application easier to develop, modify, and deploy. From the MEC point of view, microservices are natural candidate in partitioning of application processing. However, it is not a easy task to migrate an existing monolithic program to microservice-oriented one. Although edge computing is not mentioned in the paper, Mazlami et al. [72] tuckle this problem and presents a formal and semi-automatic microservice extraction method.

5.3 Resource Management

As an infrastructure, resources in a MEC system that are computation, storage, and networking bandwidth are shared among multiple users and multiple application services, in the similar way in a cloud. Moreover, in addition that MEC server generally has a smaller resource pool than that on a cloud server, it is more close to user devices and wireless access networks. This means that a sophisticated resource management is necessary to cope with the dynamism caused by user mobility and fluctuation in wireless environment. We see some novel and thought-provoking papers in this section.

Chen et al. [73] study the problem of wireless channel assignment. As the processing offload to a MEC server requires to transfer various input data over a wireless access, and the wireless channels are regarded as a shared resource among users that want offloading. The paper first shows that the computational complexity for centralized optimal solution in multi-user computation offloading problem for MEC in a multi-channel wireless interference environment is NP-hard. Then, they choose a game theoretic approach and formulate the problem as the multi-party decision making game among mobile device users. They show that the game has a Nash equilibrium and possesses the finite improvement property. In the game, each of user devices independently makes two decisions; (1) whether it offloads or execute locally, and (2) which wireless channel it requests. The decisions are based on the weighted sum of the execution time (either offload or local) and the battery consumption, and, the time for offloading communication for the offload case. The user device chooses to offload when it is beneficial to the device. As the result of repetition of independent decisions, a Nash equilibrium is reached where the maximum number of the devices benefits from offloading within the limited MEC resource. They propose a distributed algorithm that converges within a finite number of iteration. The number of iteration is at most a quadratic to the number of devices that uses a MEC server, but usually convergence is made in a time almost linear with the number of the devices. Since each iteration of the decision making only takes wireless time slot, that is 70 microseconds in LTE system, the whole assignment process takes short time in comparison with the application execution time that takes a few hundred milliseconds usually.

Kiani et al. [74] consider the problem of where to offload with a more generalized hierarchical model in which there are three tiers of MEC servers exists. They introduce the notion of *field, shallow* and *deep* cloudlets, each has different networking distances from the end devices, and different capacities for computation and storage. With multi-user setup, they formulate VM and bandwidth assignment among tiers as auction-based profit maximization problem. Though heuristics, the proposed algorithm gives very close to optimal resource allocation and works efficiently.

MEC is expected, among others, to support ultra reliable and low latency communication and processing for critical and real time applications. For them, it is important to design a MEC system not only relying on average metrics (e.g., average workload and average latency) but also considering statistical distribution of them, in other words, the upper bound of metrics or the delay bound violation probability. Liu et al. [75] investigate this problem employing *extreme value theory* [76] and Lyapunov stochastic optimization technique [77]. As the result, the method for task computation and offloading decision at the user devices, and the resource allocation at the server side are presented. Numerical analysis base on simulation shows that the proposed method is useful in estimating the delay bound violation probability and the necessary number of edge servers to accomodate the offloading requests, as well as tradeoff between power consumption and end-to-end delay at user devices.

Satria et al. [78] discuss the problem of an overloaded MEC system and its recovery, which is important issue from the view point of infrastructure operation. Because of the finite resource in edge servers and the changing workload demand from the mobile devices, a MEC server cannot avoid occasional overload (or should not, from economical view point), even by the most careful design. Rather than overload-free design, the paper proposes two recovery schemes when a MEC server is overloaded or broken. One scheme is re-offloading from the overloaded MEC server to available neighboring servers, and the other is to use user devices as ad-hoc relay nodes between the overloaded MEC server and neighboring servers. Souza et al. [79] study the similar issue of failure and recovery in MEC system, and studies two different strategies (proactive and reactive ones) on several aspects sua as service allocation time, recovery delay, and resource usage.

6. Concluding Remarks

The Multi-access Edge Computing is emerging as a totally new paradigm that will succeed the conventional client-server, or device-cloud, architecture, and expected as a future infrastacture for IoT era. In this survey, we focused on the impotrtant issues of infrastructure aspect of MEC, which are architectures, workload partitioning and tradeoff, and resource management. There are many other good surveys for MEC and edge computing from various points of view, such as Refs. [80], [81] and [82], among others. The security implication of MEC, which we did not mentioned, are surveyed in Ref. [83]. Naturally, the MEC concept relates to a broad range of research areas, not limited to technical aspects but also business and economical ones. While some of the necessary technologies can be brought from long and well investigated fields, such as distributed processing, cloud technologies, virtual machines, etc., there are a lot of research challenges and opportunities to be investigated. Although the growing interest in MEC from both academia and industry in recent years, research in the area of MEC-specific issues is still in the early stage; to name few, how to partition an application logic for MEC automatically or without too much programmers' intervention, how to relocate or handover an ongoing processing among edge servers and between an edge server and a cloud to cope with the user device mobility or change in workload, or methodology for edge-oriented software design and development, among others. Both extensive and intensive efforts are greatly desired.

References

[1] International Telecommunication Union Telecommunication Standardization Sector: The Tactile Internet, *ITU-T Technology Watch Report* (2014) (online), available from ⟨https://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000230001PDFE.pdf⟩ (accessed 2017-10-17).

[2] Gelsinger, P.P.: Microprocessors for the new millennium: Challenges, opportunities, and new frontiers, *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*), pp.22–25 (2001).

[3] Zhang, Z., Xie, X., Lu, B. and Lin, S.: Enabling rich content service on the edge: Opportunities and challenges, *Microsoft Technical Report*, MSR-TR-2002-71 (2002) (online), available from ⟨https://www.microsoft.com/en-us/research/publication/enabling-rich-content-service-on-the-edge-opportunities-and-challenges/⟩ (accessed 2017-10-17).

[4] Davis, A., Parikh, J. and Weihl, W.E.: Edgecomputing: extending enterprise applications to the edge of the internet, *Proc. 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pp.180–187, ACM (2004).

[5] Satyanarayanan, M., Bahl, P., Caceres, R. and Davies, N.: The case for

vm-based cloudlets in mobile computing, *IEEE Pervasive Computing*, Vol.8, No.4, pp.14–23 (2009).

[6] Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Chandra, R. and Bahl, P.: MAUI: Making smartphones last longer with code offload, *Proc. 8th International Conference on Mobile Systems, Applications, and Services*, pp.49–62, ACM (2010).

[7] Chun, B.-G., Ihm, S., Maniatis, P., Naik, M. and Patti, A.: Clonecloud: Elastic execution between mobile device and cloud, *Proc. 6th Conference on Computer Systems*, pp.301–314, ACM (2011).

[8] Heo, J., Terada, K., Toyama, M., Kurumatani, S. and Chen, E.Y.: User demand prediction from application usage pattern in virtual smartphone, *2nd International Conference on Cloud Computing Technology and Science* (*CloudCom*), pp.449–455, IEEE (2010).

[9] European Telecommunications Standards Institute: Multi-access Edge Computing, ETSI ISG MEC (web page) (online), available from ⟨http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing/⟩ (accessed 2017-10-17).

[10] European Telecommunications Standards Institute: Mobile-edge computing - Introductory Technical White Paper, *ETSI ISG MEC* (2014) (online), available from ⟨https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf⟩ (accessed 2017-10-17).

[11] Reznik, A., Arora, R., Cannon, M., Cominardi, L., Featherstone, W., Frazao, R., Giust, F., Kekki, S., Li, A., Sabella, D., Turyagyenda, C. and Zheng, Z.: Developing Software for Multi-Access Edge Computing, *ETSI White Paper*, No.20 (2017) (online), available from ⟨http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp20_MEC_SoftwareDevelopment_FINAL.pdf⟩ (accessed 2017-10-17).

[12] European Telecommunications Standards Institute: Mobile-edge computing (MEC); Service Scenarios, *ETSI Group Specification*, GS MEC-IEG 004, V1.1.1 (2015) (online), available from ⟨http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf⟩ (accessed 2017-10-17).

[13] European Telecommunications Standards Institute: Mobile-edge computing (MEC); Technical Requirements, *ETSI Group Specification*, GS MEC 002, V1.1.1 (2016) (online), available from ⟨http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf⟩ (accessed 2017-10-17).

[14] European Telecommunications Standards Institute: Mobile-edge computing (MEC); Framework and Reference Architecture, *ETSI Group Specification*, GS MEC 003, V1.1.1 (2016) (online), available from ⟨http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf⟩ (accessed 2017-10-17).

[15] Goudarzi, H. and Pedram, M.: Multi-dimensional SLA-Based Resource Allocation for Multi-tier Cloud Computing Systems, *IEEE 4th International Conference on Cloud Computing*, pp.324–331 (2011).

[16] Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M. and Tantawi, A.: An Analytical Model for Multi-tier Internet Services and Its Applications, *SIGMETRICS Perform. Eval. Rev.*, Vol.33, No.1, pp.291–302 (2005).

[17] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S.: Fog computing and its role in the internet of things, *Proc. 1st Edition of the MCC Workshop on Mobile Cloud Computing*, pp.13–16, ACM (2012).

[18] Open Edge Computing Initiative: Living Edge Lab, Open Edge Computing (web page) (online), available from ⟨http://openedgecomputing.org/lel.html⟩ (accessed 2017-10-17).

[19] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L.: Edge computing: Vision and challenges, *IEEE Internet of Things Journal*, Vol.3, No.5, pp.637–646 (2016).

[20] Takahashi, N., Tanaka, H. and Kawamura, R.: Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated web browsing, *3rd International Conference on Mobile Cloud Computing, Services, and Engineering* (*MobileCloud*), pp.233–234, IEEE (2015).

[21] Wang, C.C., Lin, Z.N., Yang, S.R. and Lin, P.: Mobile edge computing-enabled channel-aware video streaming for 4G LTE, *2017 13th International Wireless Communications and Mobile Computing Conference* (*IWCMC*), pp.564–569 (2017).

[22] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (coordinator): SUPERFLUIDITY project, SUPERFLUIDITY Project (web page) (online), available from ⟨http://superfluidity.eu/⟩ (accessed 2017-10-17).

[23] Salsano, S., Chiaraviglio, L., Blefari-Melazzi, N., Parada, C., Fontes, F., Mekuria, R. and Griffioen, D.: Toward superfluid deployment of virtual functions: Exploiting mobile edge computing for video streaming, *Proc. 1st International Workshop on Softwarized Infrastructures for 5G and Fog Computing*, Genoa, Italy, pp.4–8 (2017).

[24] Docker, Inc.: What is Docker, Docker Project (web page) (online), available from ⟨https://www.docker.com/what-docker⟩ (accessed 2017-10-17).

[25] Tran, T.X., Pandey, P., Hajisami, A. and Pompili, D.: Collaborative multi-bitrate video caching and processing in Mobile-Edge Computing networks, *13th Annual Conference on Wireless On-demand Network Systems and Services* (*WONS*), pp.165–172 (2017).

[26] Anjum, A., Abdullah, T., Tariq, M., Baltaci, Y. and Antonopoulos, N.: Video Stream Analysis in Clouds: An Object Detection and Classification Framework for High Performance Video Analytics, *IEEE Trans. Cloud Computing*, Vol.PP, No.99 (2016).

[27] Satyanarayanan, M., Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Hu, W. and Amos, B.: Edge analytics in the internet of things, *IEEE Pervasive Computing*, Vol.14, No.2, pp.24–31 (2015).

[28] Hewlett Packard Enterprise: Edge Video Analytics, Hewlett Packard Enterprise (web page) (online), available from ⟨http://www.hpe.com/support/EL-EVA⟩ (accessed 2017-10-17).

[29] Intel: Video Analytics at the Edge, Intel (web page) (online), available from ⟨http://www.intel.com/content/www/us/en/internet-of-things/videos/iot-intel-surveillance-video.html⟩ (accessed 2017-10-17).

[30] Verbelen, T., Simoens, P., De Turck, F. and Dhoedt, B.: Cloudlets: Bringing the cloud to the mobile user, *Proc. 3rd ACM Workshop on Mobile Cloud Computing and Services*, pp.29–36, ACM (2012).

[31] Chen, Z., Jiang, L., Hu, W., Ha, K., Amos, B., Pillai, P., Hauptmann, A. and Satyanarayanan, M.: Early implementation experience with wearable cognitive assistance applications, *Proc. 2015 workshop on Wearable Systems and Applications*, pp.33–38, ACM (2015).

[32] Dolezal, J., Becvar, Z. and Zeman, T.: Performance evaluation of computation offloading from mobile device to the edge of mobile network, *IEEE Conference on Standards for Communications and Networking* (*CSCN*), pp.1–7 (2016).

[33] Mangiante, S., Klas, G., Navon, A., GuanHua, Z., Ran, J. and Silva, M.D.: VR is on the Edge: How to Deliver 360° Videos in Mobile Networks, *Proc. Workshop on Virtual Reality and Augmented Reality Network*, pp.30–35, ACM (2017).

[34] 3rd Generation Partnership Project: 3GPP, 3rd Generation Partnership Project (web page) (online), available from ⟨http://www.3gpp.org/⟩ (accessed 2017-10-17).

[35] Brito, M.S.D., Hoque, S., Steinke, R. and Willner, A.: Towards Programmable Fog Nodes in Smart Factories, *IEEE 1st International Workshops on Foundations and Applications of Self* Systems* (*FAS*W*), pp.236–241 (2016).

[36] Soldatos, J., Gusmeroli, S., Malo, P. and Di Orio, G.: Internet of Things Applications in Future Manufacturing, *Digitising Industry-Internet of Things Connecting the Physical, Digital and Virtual Worlds* (2016).

[37] Intel: Using Big Data in Manufacturing at Intel's Smart Factories, *Intel White Paper* (2016) (online), available from ⟨https://www.intel.com/content/dam/www/public/us/en/documents/best-practices/using-big-data-in-manufacturing-at-intels-smart-factories-paper.pdf⟩ (accessed 2017-10-17).

[38] IBM: IoT edge analytics is transforming manufacturing, IBM (web page) (online), available from ⟨https://www.ibm.com/blogs/internet-of-things/smart-manufacturing-edge-analytics/⟩ (accessed 2017-10-17).

[39] FANUC: Agreement on cooperation for the speedy establishment and service launch of the FIELD system optimizing manufacturing production with IoT, FANUC (press release) (2016) (online), available from ⟨http://www.fanuc.co.jp/en/profile/pr/newsrelease/notice20160728.html⟩ (accessed 2017-10-17).

[40] Amadeo, M., Campolo, C. and Molinaro, A.: Information-centric networking for connected vehicles: A survey and future perspectives, *IEEE Communications Magazine*, Vol.54, No.2, pp.98–104 (2016).

[41] Grewe, D., Wagner, M., Arumaithurai, M., Psaras, I. and Kutscher, D.: Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions, *Proc. Workshop on Mobile Edge Communications*, pp.7–12, ACM (2017).

[42] 5G Automotive Association: The Case for Cellular V2X for Safety and Cooperative Driving (white paper), *5GAA White Paper* (2016) (online), available from ⟨http://5gaa.org/pdfs/5GAA-whitepaper-23-Nov-2016.pdf⟩ (accessed 2017-10-17).

[43] 5G Infrastructure Public Private Partnership: 5G Automotive Vision, *5GPPP White Paper* (2015) (online), available from ⟨https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf⟩ (accessed 2017-10-17).

[44] Denso, Ericsson, Intel, NTT, NTT Docomo, Toyota Motor, and Toyota ITC: Industry leaders to form consortium for network and computing infrastructure of automotive big data, Toyota Motor et al. (press release) (2017) (online), available from ⟨http://newsroom.toyota.co.jp/en/detail/18135029⟩ (accessed 2017-10-17).

[45] Behrisch, M., Bieker, L., Erdmann, J. and Krajzewicz, D.: SUMO–simulation of urban mobility: An overview, *Proc. SIMUL 2011, The 3rd International Conference on Advances in System Simulation*, ThinkMind (2011).

[46] Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L.: Recent development and applications of SUMO-Simulation of Urban MObility, *International Journal On Advances in Systems and Measurements*, Vol.5, No.3&4, pp.128–138 (2012).
[47] Villari, M., Fazio, M., Dustdar, S., Rana, O. and Ranjan, R.: Osmotic computing: A new paradigm for edge/cloud integration, *IEEE Cloud Computing*, Vol.3, No.6, pp.76–83 (2016).
[48] Kubernetes Project: Kubernetes - Producton-Grade Container Orchestration, Kubernetes Project (web page) (online), available from ⟨https://kubernetes.io/⟩ (accessed 2017-10-31).
[49] Sapienza, M., Guardo, E., Cavallo, M., La Torre, G., Leombruno, G. and Tomarchio, O.: Solving critical events through mobile edge computing: An approach for smart cities, *International Conference on Smart Computing* (*SMARTCOMP*), IEEE (2016).
[50] Sun, X. and Ansari, N.: EdgeIoT: Mobile edge computing for the internet of things, *IEEE Communications Magazine*, Vol.54, No.12, pp.22–29 (2016).
[51] European Telecommunications Standards Institute: Network Functions Virtualization, ETSI ISG NFV (web page) (online), available from ⟨http://www.etsi.org/technologies-clusters/technologies/nfv⟩ (accessed 2017-10-17).
[52] Open Fog Consortium: Open Fog Consortium, Open Fog Consortium (web page) (online), available from ⟨https://www.openfogconsortium.org/⟩ (accessed 2017-10-17).
[53] Open Edge Computing Initiative: Open Egde Computing, Open Edge Computing (web page) (online), available from ⟨http://openedgecomputing.org/⟩ (accessed 2017-10-17).
[54] Telecom Infra Project: Edge Computing, Telecom Infra Project (web page) (online), available from ⟨https://telecominfraproject.com/project/access-projects/edge-computing/⟩ (accessed 2017-10-17).
[55] Edge Computing Consortium: Edge Computing Consortium, Edge Computing Consortium (web page) (online), available from ⟨http://en.ecconsortium.org/⟩ (accessed 2017-10-17).
[56] Tran, T.X., Hajisami, A., Pandey, P. and Pompili, D.: Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges, *IEEE Communications Magazine*, Vol.55, No.4, pp.54–61 (2017).
[57] International Telecommunication Union Radiocommunication Sector: IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond, *ITU-R Recommendation*, M.2083-0 (2015) (online), available from ⟨https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf⟩ (accessed 2017-10-17).
[58] SESAME project: Small cEllS coordinAtion for Multi-tenancy and Edge services (SESAME), SESAME Project (web page) (online), available from ⟨http://www.sesame-h2020-5g-ppp.eu/⟩ (accessed 2017-10-17).
[59] SESAME project: SESAME: An innovative multi-operator enabled Small Cell based infrastructure that integrates a virtualised execution platform for deploying Virtual Network Functions, *SESAME Project 2nd White Paper* (2017) (online), available from ⟨http://www.sesame-h2020-5g-ppp.eu/Portals/0/Dissemination/SESAME%20Second%20White%20Paper_final.pdf⟩ (accessed 2017-10-17).
[60] Cziva, R. and Pezaros, D.P.: Container Network Functions: Bringing NFV to the Network Edge, *IEEE Communications Magazine*, Vol.55, No.6, pp.24–31 (2017).
[61] Carella, G.A., Pauls, M., Magedanz, T., Cilloni, M., Bellavista, P. and Foschini, L.: Prototyping nfv-based multi-access edge computing in 5G ready networks with open baton, *2017 IEEE Conference on Network Softwarization* (*NetSoft*) pp.1–4, IEEE (2017).
[62] Open Baton Project: Open Baton, an extensible and customizable NFV MANO-compliant framework, Open Baton Project (web page) (online), available from ⟨http://openbaton.github.io/⟩ (accessed 2017-10-17).
[63] Rimal, B.P., Van, D.P. and Maier, M.: Mobile Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era, *IEEE Communications Magazine*, Vol.55, No.2, pp.192–200 (2017).
[64] Orsini, G., Bade, D. and Lamersdorf, W.: Cloudaware: A context-adaptive middleware for mobile edge and cloud computing applications, *IEEE International Workshops on Foundations and Applications of Self* Systems*, pp.216–221, IEEE (2016).
[65] Laurila, J.K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.M.T., Dousse, O., Eberle, J. and Miettinen, M.: From big smartphone data to worldwide research: The mobile data challenge, *Pervasive and Mobile Computing*, Vol.9, No.6, pp.752–771 (2013).
[66] Flinn, J.: Cyber foraging: Bridging mobile and cloud computing, *Synthesis Lectures on Mobile and Pervasive Computing*, Vol.7, No.2, pp.1–103 (2012).
[67] Hu, W., Gao, Y., Ha, K., Wang, J., Amos, B., Chen, Z., Pillai, P. and Satyanarayanan, M.: Quantifying the impact of edge computing on mobile applications, *Proc. 7th ACM SIGOPS Asia-Pacific Workshop on Systems*, p.5, ACM (2016).
[68] Gordon, M.S., Jamshidi, D.A., Mahlke, S.A., Mao, Z.M. and Chen, X.: COMET: Code Offload by Migrating Execution Transparently, *OSDI*, Vol.12, pp.93–106 (2012).
[69] Wang, Y., Sheng, M., Wang, X., Wang, L. and Li, J.: Mobile-edge computing: Partial computation offloading using dynamic voltage scaling, *IEEE Trans. Commun.*, Vol.64, No.10, pp.4268–4282 (2016).
[70] Mao, Y., Zhang, J., Song, S. and Letaief, K.B.: Power-delay tradeoff in multi-user mobile-edge computing systems, *2016 IEEE Global Communications Conference* (*GLOBECOM*), pp.1–6, IEEE (2016).
[71] Mao, Y., Zhang, J. and Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices, *IEEE Journal on Selected Areas in Communications*, Vol.34, No.12, pp.3590–3605 (2016).
[72] Mazlami, G., Cito, J. and Leitner, P.: Extraction of Microservices from Monolithic Software Architectures, *2017 IEEE International Conference on Web Services* (*ICWS*), pp.524–531, IEEE (2017).
[73] Chen, X., Jiao, L., Li, W. and Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing, *IEEE/ACM Trans. Netw.*, Vol.24, No.5, pp.2795–2808 (2016).
[74] Kiani, A. and Ansari, N.: Towards Hierarchical Mobile Edge Computing: An Auction-Based Profit Maximization Approach, *IEEE Internet of Things Journal*, Vol.PP, No.99 (2017).
[75] Liu, C.-F., Bennis, M. and Poor, H.V.: Latency and Reliability-Aware Task Offloading and Resource Allocation for Mobile Edge Computing, arXiv preprint arXiv:1710.00590 (2017).
[76] Coles, S., Bawa, J., Trenner, L. and Dorazio, P.: *An introduction to statistical modeling of extreme values*, Vol.208, Springer (2001).
[77] Neely, M.J.: Stochastic network optimization with application to communication and queueing systems, *Synthesis Lectures on Communication Networks*, Vol.3, No.1, pp.1–211 (2010).
[78] Satria, D., Park, D. and Jo, M.: Recovery for overloaded mobile edge computing, *Future Generation Computer Systems*, Vol.70, pp.138–147 (2017).
[79] Souza, V.B., Masip-Bruin, X., Marín-Tordera, E., Ramírez, W. and Sánchez-López, S.: Proactive vs reactive failure recovery assessment in combined Fog-to-Cloud (F2C) systems, *IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks* (*CAMAD*) (2017).
[80] Ahmed, A. and Ahmed, E.: A survey on mobile edge computing, *2016 10th International Conference on Intelligent Systems and Control* (*ISCO*), IEEE (2016).
[81] Mao, Y., You, C., Zhang, J., Huang, K. and Letaief, K.B.: Mobile edge computing: Survey and research outlook, arXiv preprint arXiv:1701.01090 (2017).
[82] Mahmud, R., Kotagiri, R. and Buyya, R.: Fog computing: A taxonomy, survey and future directions, *Internet of Everything*, pp.103–130, Springer (2017).
[83] Roman, R., Lopez, J. and Mambo, M.: Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, *Future Generation Computer Systems*, Vol.78, pp.680–698 (2016).

Hiroyuki Tanaka is a Senior Research Engineer, Supervisor in NTT Network Innovation Labs. He received B.E degree from Kyushu University, and M.E. degree from Nara Institute of Science and Technology, in 1993 and 1995, respectively. He joined NTT (Nippon Telegraph and Telephone Corp.) in 1995. His current interests include design and implementation of future computing systems.

Masahiro Yoshida received Ph.D. degree (2013) from the University of Tokyo, Japan. He was a research fellowship for young scientists at Japan Society for the Promotion of Science (JSPS) in 2011 and 2013. He is currently a research engineer at NTT Network Innovation Laboratories, Nippon Telegraph and Telephone Corporation. His research interests include MEC, NFV and SDN.

Koya Mori is a senior research engineer in NTT Network Innovation Laboratories. After joining the NTT corporation in 2004, he worked for research and development of an IoT application for a long time based on software technologies such as the OSGi, the OpenStack and the Edge Computing.

Noriyuki Takahashi received B.E. and M.E. degrees in information engineering from Kyoto University, in 1990 and 1992, respectively. He joined NTT laboratories in 1992. He has been in the NTT Network Innovation Labs. since 2001. His current interests include routing mechanisms, adaptive networking systems, and architecture design of future networks. He is a member of IEICE and IEEE.