Adaptive Service Offloading for Revenue Maximization in Mobile Edge Computing With Delay-Constraint

Amit Samanta, *Member, IEEE*, and Zheng Chang, *Senior Member, IEEE*

***Abstract*—Mobile edge computing (MEC) is an important and effective platform to offload the computational services of modern mobile applications, and has gained tremendous attention from various research communities. For delay and resource constrained mobile devices, the important issues include: 1) minimization of the service latency; 2) optimal revenue maximization; and 3) high quality-of-service requirement to offload the computational service offloading. To address the above issues, an adaptive service offloading scheme is designed to provide the maximum revenue and service utilization to MEC. Unlike most of the existing works, we consider both the delay-tolerant and delay-constraint services in order to achieve the optimized service latency and revenue. Furthermore, we consider the different priorities to prioritize the edge services for optimal service offloading. We formulate the proposed scheme mathematically. Simulation results are presented to demonstrate the effectiveness of the proposed adaptive service offloading scheme over other existing state-of-the-art solutions, in terms of service latency, utility value, revenue, and utilization.**

***Index Terms*—Adaptive service offloading, mobile edge computing (MEC), performance analysis, revenue maximization.**

I. Introduction

In recent times cloud platform has become very important platform for modern day mobile applications to support Tactile Internet infrastructure [1], [2]. Generally, the conventional standalone applications execute their services fully at the mobile device, whereas cloud applications execute their services at the cloud, as they composed of multiple components. Hence, one of the components of cloud applications execute at the cloud and another component running on the mobile device, jointly they establish an application available to mobile users [3]. Such mobile cloud applications require high data processing, infrastructures, storage capability that may not be fulfilled on the standalone mobile devices, thus it is necessary to run part of the application in the cloud. Generally, the cloud servers are placed in a centralized data centers. However, the fundamental problem with cloud computing is the higher service latency and intermittent connectivity between the mobile devices and cloud servers, which may not be able to satisfy the real-time services of different emerging applications, such as augmented reality and online traffic monitoring systems. To solve this issues, mobile edge computing (MEC) has become an important and effective platform for different real-time mobile applications [4]–[6]. The fundamental objective of MEC is to design a small-scale cloud platform deployed at the edge of the network, where different mobile edge devices execute their computational services of different applications, like traffic monitoring and healthcare [7]–[14]. Such platforms are placed nearer to the proximity of users to provide seamless and low-latency access to edge services.

In this paper, we study the revenue maximization problem for computational service offloading in MEC platform from a edge service provider's point of view. To execute the computational services, edge devices submit computational offloading requests, containing source and destination addresses and offloading time intervals, to the edge service provider. The edge service provider designs a revenue maximization problem to specify the resource charges to potential edge devices. Edge devices response to the charges by choosing a computational offloading rate to transmit data over the network. This is one of the preliminary work on MEC with the objective of maximizing total revenue, which measures the aggregated service utilization of edge devices. This objective may be in the interest of both edge service providers and edge devices, where the edge service provider and devices want to extract more revenue to maximize the profit level. Thus, our focus here is on designing an adaptive service offloading scheme for revenue maximization. The main contributions of this paper are discussed below.

1) We propose an adaptive service offloading scheme for MEC to maximize the total revenue, while maintaining total utility value of the network. We also present an optimal revenue optimization problem to maximize the profit level of both edge devices and servers for MEC.
2) We consider the delay-sensitive and delay-tolerant edge services in designing the adaptive service offloading algorithm. We also consider the optimal demand of edge devices for efficient service offloading. Thus, we estimate the total demand of edge devices effectively in order to minimize the service latency.
3) Simulation results demonstrate that our algorithm can effectively offload the computational services from edge

Manuscript received September 20, 2018; revised December 5, 2018; accepted December 25, 2018. Date of publication January 10, 2019; date of current version May 8, 2019. *(Corresponding author: Zheng Chang.)*

A. Samanta is with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, India (e-mail: amit.samanta049@gmail.com).

Z. Chang is with the Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland (e-mail: zheng.chang@jyu.fi).

Digital Object Identifier 10.1109/JIOT.2019.2892398

devices to edge servers. The results also show that the proposed scheme provides higher service utilization while minimizing total service latency. It also yields the best performance, in terms of utility value and total revenue, under different performance settings, compared with other solutions.

The rest of this paper is organized as follows. Section II describes the related work. In Section III, we present the system model for MEC. Section IV describes an adaptive service offloading scheme for MEC, in particular, our adaptive service offloading scheme is designed to provide maximum revenue. Section VI conducts extensive simulations to validate our proposed scheme, and Section VII concludes this paper.

II. Related Work

The problem of computational service offloading with optimal resource and delay is a challenging task for MEC. Over the years, only a few researchers have addressed some of the important issues related to this problem. Mao *et al.* [15] proposed a dynamic computation offloading scheme for MEC with energy-harvesting edge devices. You *et al.* [16] proposed an energy-efficient resource allocation scheme for service offloading in MEC. Ko *et al.* [17] proposed a live prefetching scheme for computational service offloading of edge devices. Zhao *et al.* [18] proposed a task scheduling and resource allocation scheme for delay-bound MEC platform. Zhang *et al.* [19] an auction-based service provider selection scheme for MEC. Dinh *et al.* [20] proposed task allocation and frequency scaling for optimal service offloading in MEC. Ti and Le [21] proposed computational resource allocation scheme for service offloading in edge clouds. Li *et al.* [22] proposed data-analysis for IoT applications in MEC to leverage renewable energy. Shekhar and Gokhale [23] proposed a dynamic resource management scheme for mobile edge clouds. Reiter *et al.* [24] proposed a hybrid edge computing platform to unleash the full potential of MEC platform for IoT applications. Chang *et al.* [25] proposed energy-efficient optimization framework for computation offloading in fog computing system. Liu *et al.* [26] proposed a multiobjective optimization problem for computation offloading in fog computing environment. Wu *et al.* [27] studied the nonorthogonal multiple access-enabled multiaccess MEC to minimize the overall-delay of the mobile users, by jointly optimizing the users' offloaded workloads and the NOMA transmission-time. Samanta and Li [28] proposed an optimal economical framework for MEC. Samanta *et al.* [29] proposed a latency-oblivious distributed task scheduling algorithm for MEC. Samanta and Li [30] proposed a latency-oblivious service offloading scheme for MEC.

In summary, most for the existing studies [15]–[30] mainly focus on the energy-efficient and resource-efficient computational service offloading scheme in MEC. They did not propose any revenue maximization problem for service offloading in MEC to provide the optimal profits to both edge devices and service provider. Thus, for delay- and resource-constraint edge devices, the service offloading scheme is very important in order to provide fair resources and optimal latency to edge devices. This motivates us to design an adaptive service offloading scheme for revenue maximization in MEC platform.

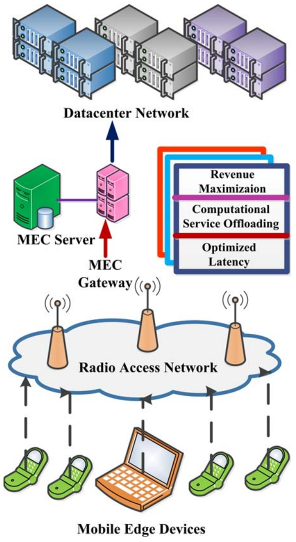


Fig. 1. Adaptive service offloading in MEC.

III. System Model

Without loss of generality, we assume there are N edge devices denoted by, $\mathcal{ED} = \{\mathrm{ED}_1, \mathrm{ED}_2, \ldots, \mathrm{ED}_N\}$, existing in order to offload the computational services, as shown in Fig. 1. From the figure, we can see that a set of edge devices are trying to offload their computational services with minimum service delay thorough a backhaul radio access network. Each of the edge device has different kind of services denoted by $\mathbb{S} = \{S_1, S_2, \ldots, S_K\}$ and they belong to different real-time mobile applications (i.e., self-driving car, augmented-reality, traffic monitoring system etc.). The edge devices offload their computational services to edge servers denoted by, $\mathcal{ES} = \{\mathrm{ES}_1, \mathrm{ES}_2, \ldots, \mathrm{ES}_M\}$, where each edge server has a certain computational capacity. The owners of the edge servers is considered to be edge service providers denoted by $\mathcal{SP} = \{\mathrm{SP}_1, \mathrm{SP}_2, \ldots, \mathrm{SP}_O\}$, hence they can ask for prices to execute their services. In this scheme, the computational services S_{arr} arrive at edge servers by uniform distribution and the offload of computational services organized according to their priorities. However, the edge devices require fair amount of bandwidths to offload their computational services effectively. We assume that the edge device ED_i has a maximum $\mathbb{B}_i^{mx}$ and a minimum $\mathbb{B}_i^{my}$ bandwidth requirements to offload the computational services to edge servers.

Fundamentally, the edge devices has very limited power to offload their computational services, therefore the service offloading mechanism is very important to minimize the energy consumption rate of edge devices. Here, we consider the initial energy of a edge device ED_i is $\mathcal{E}_{\mathrm{ini}}^i$. Along with energy consumption, it is necessary to minimize the service offloading price in order to maximize the revenue of edge service providers and devices. Here, the total service offloading price for a edge device ED_i is denoted by $\mathbb{P}_{\mathrm{off}}^t$. Therefore, we propose an adaptive service offloading scheme for MEC to minimize the service latency and offloading price. Thereafter, we propose an adaptive service offloading algorithm to maximize the revenue of edge devices.

IV. Adaptive Service Offloading

Due to heavy network load and congestion, the service offloading latency and price increases in the network, which inherently minimizes the service utilization in MEC. In order to improve the service utilization and maximize the revenue of edge devices, here we discuss an adaptive service offloading scheme for MEC. At first, we need to estimate the total service latency encountered by edge devices in the network, while offloading the computational services. Later, we propose a utility maximization problem, while taking into consideration of service priority of edge devices.

A. Approximation of Service Latency

The total service latency encountered by mobile edge devices is estimated based on the total *service execution* and *service offloading* latency. They are explained in details below.

1) *Service Execution Latency:* The service execution latency $\mathbb{D}^t_{\mathrm{EL}}$ of edge device ED_i is depended on the total number of CPU cycles required to execute service S_i and the local service computing capacity distributed to service S_i by edge server. We have

$$\mathbb{D}^t_{\mathrm{EL}} = \left[\frac{\mathcal{G}_i}{F^{\mathrm{lo}}_i} + \mathbb{W}^t_i\right] \tag{1}$$

where $\mathcal{G}_i$ denotes the total number of CPU cycles required to execute service S_i, F^{lo}_i denotes local service computing capacity distributed to service S_i by edge server, and $\mathbb{W}^t_i$ denotes the initial waiting time to get the required number of CPU cycles required to execute service.

2) *Service Offloading Latency:* The service offloading latency $\mathbb{D}^t_{\mathrm{off}}$ is directly proportional to the total waiting time to offload the computational services on edge devices. It is defined as

$$\mathbb{D}^t_{\mathrm{off}} = \frac{\mathbb{J}^t_i}{q^t_i} \tag{2}$$

where $\mathbb{J}^t_i$ and q^t_i denote the total service length and time to execute the service S_i at time t, respectively.

Hence, the total estimated service latency $\mathbb{D}^t_{to}$ for edge device ED_i is the addition of both service execution latency $\mathbb{D}^t_{\mathrm{EL}}$ and service offloading latency $\mathbb{D}^t_{\mathrm{off}}$, which is defined as

$$\mathbb{D}^t_{to} = \mathbb{D}^t_{\mathrm{EL}} + \mathbb{D}^t_{\mathrm{off}} = \left[\frac{\mathcal{G}_i}{F^{\mathrm{lo}}_i} + \mathbb{W}^t_i\right] + \frac{\mathbb{J}^t_i}{q^t_i}. \tag{3}$$

B. Optimal Revenue Maximization

After estimation of total service latency, we now model an adaptive service offloading scheme for edge devices to maximize the revenue and service utilization. Suppose, we assume that the computational services require $\mathbb{T}$ slots to offload them efficiently to edge servers. Here, we consider a time frame with different time slots. We describe the length of time slot T and index of time-slot by t, where $t \in \mathbb{T} = \{1, 2, \ldots, \}$. In a time-slot, if more than one edge device choose a particular channel to offloading their services, then we have used the carrier sense multiple access mechanism to overcome the possible collisions in the network. To maximize the revenue of devices, it is necessary to estimate the total demand and price for computational service offloading.

Definition 1: The demand profile $\mathcal{Z}$ of edge devices is denoted by a set of demand variables, $\mathcal{Z} = \{\mathbb{Z}_1, \mathbb{Z}_2, \ldots, \mathbb{Z}_{\mathbb{N}}\}$. The demand variable of a edge device is dependent on the total bandwidth requirement and service execution time. We have

$$\mathcal{Z}_i = \{\mathbb{B}^{\mathrm{req}}_i, t_i\} \tag{4}$$

where $\mathbb{B}^{\mathrm{req}}_i$ and t_i denote the bandwidth and execution time required to offload the computational services.

Definition 2: The average bandwidth requirement of edge devices is estimated using a classical exponential moving average technique. It is defined as

$$\mathbb{B}^{\mathrm{req}}_i = \bar{\lambda_i}(t) = \alpha\bar{\lambda_i}(t-1) + (1-\alpha)\lambda_i(t) \tag{5}$$

where $\bar{\lambda_i}(t)$ denotes the average bandwidth requirement (i.e., average reward) at time t, α denotes the exponential moving average rate, and $\bar{\lambda_i}(t-1)$ and $\lambda_i(t)$ denote the average bandwidth requirement at time $t-1$ and absolute bandwidth requirement at time t.

Definition 3: The service offloading rate of edge device ED_i is dependent on the data size of computational services and total number of time-slots required for offloading, which is mathematically defined as

$$\mathcal{H}_i(\mathcal{Y}, t) = \mathbb{V}_i\zeta_i(\mathcal{Y}, t) \tag{6}$$

where $\mathbb{V}_i$ denotes the data size of computational services and $\zeta_i(\mathcal{Y}, t)$ indicates whether edge device ED_i successfully gets a time-slot t with probability $\mathcal{Y}$, which is defined as

$$\zeta_i(\mathcal{Y}, t) = \begin{cases} 1, & \text{edge device } \mathrm{ED}_i \text{ successfully gets a time-slot} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 4: The probability $\mathcal{Y}$ of getting a time-slot t to offload the computational services is defined as

$$\mathcal{Y} = \begin{cases} \prod(1-\mathbb{T}_{S_i}), & \text{if } \mathbb{T}_{S_i} > \mathbb{T}_{\mathrm{th}} \\ 1-\prod(1-\mathbb{T}_{S_i}), & \text{otherwise} \end{cases} \tag{7}$$

where $\mathbb{T}_{S_i}$ and $\mathbb{T}_{\mathrm{th}}$ denote the deadline of service S_i and threshold deadline, respectively.

Definition 5: The data size of a computational service for edge device ED_i is dependent on the service offloading rate and offloading time. Mathematically

$$\mathcal{X}_i = \mathcal{H}_i(\mathcal{Y}, t)t^{\mathrm{off}}_i \tag{8}$$

where t^{off}_i denotes the offloading time and $\mathcal{H}_i(\mathcal{Y}, t)$ denotes the service offloading rate.

Definition 6: The quality-of-service (QoS)-level is defined as the ratio of total number of computational services offloaded successfully to edge servers and the total service latency of edge devices. It is defined as

$$\mathbb{Q}_i = \frac{\mathcal{X}_i}{\sum_{i \in N} \sum_{t \in \mathbb{T}} \mathbb{D}_{to}^t} \tag{9}$$

where $\mathcal{X}_i$ denotes data size of a computational service for edge device ED_i and $\mathbb{D}_{to}^t$ denotes the total estimated service latency for edge device ED_i.

Definition 7: The edge service provider charge a price for computational service offloading, which is dependent on the inflow and outflow services. Mathematically

$$\mathcal{P}_i^{\text{off}} = \sum_{i \in N} \sum_{j \in M} \sum_{t \in \mathbb{T}} \Big(\gamma \mathcal{A}_{ij}^{\text{in}}(t) x_{if}^t + \psi \mathcal{B}_{ji}^{\text{out}}(t) y_{of}^t \Big) \tag{10}$$

where γ and ψ denote the unit price for both in- and outflow services. $\mathcal{A}_{ij}^{\text{in}}(t)$ and $\mathcal{B}_{ji}^{\text{out}}(t)$ denote the in- and out-flow service offloading at time t. x_{if}^t and y_{of}^t denote the unit service offloading price for up- and down-link at time t, respectively.

Definition 8: The edge devices contentiously request for bandwidths to execute their computational services. In order to provide the fair bandwidths, the services are mapped to different edge servers in a datacenter. To map the edge servers, the edge service provider charge a price to execute the services with limited delay. In addition to mapping cost, the edge servers charge other prices for edge servers management. The server management price, $\mathcal{P}_i^{\text{sm}}$, is dependent on the mapping price $\mathcal{P}_i^{\text{map}}$, initial sever development price $\mathcal{P}_i^{\text{sd}}$, and operational price $\mathcal{P}_i^{\text{op}}$, which is defined as

$$\mathcal{P}_i^{\text{sm}} = \mathcal{P}_i^{\text{map}} + \mathcal{P}_i^{\text{sd}} + \mathcal{P}_i^{\text{op}}. \tag{11}$$

The price changes to edge devices by edge service provider also incorporates the price of virtual machine (VM) creation, management, and migration. Thus, the total price of VM configuration and reconfiguration is expressed as

$$\mathcal{P}_i^{\text{vm}} = \mathcal{P}_i^{\text{vm}_c\text{re}} + \mathcal{P}_i^{\text{vm}_m ang} \tag{12}$$

where $\mathcal{P}_i^{\text{vm}_c\text{re}}$ and $\mathcal{P}_i^{\text{vm}_m ang}$ denote the unit VM creation price and management price, respectively.

The total price $\mathcal{P}_i^{\text{tot}}$ charge by edge service provider is defined as, $\mathcal{P}_i^{\text{tot}} = \mathcal{P}_i^{\text{off}} + \mathcal{P}_i^{\text{sm}} + \mathcal{P}_i^{\text{vm}}$.

V. Offloading Decision Framework

Using Definitions 2–8, we formulate net utility $\mathcal{U}_i$ for computational service offloading from edge devices to servers, which is expressed as

$$\mathbb{U}_i = \left(\Delta_1 \mathcal{H}_i(\mathcal{Y}, t) \mathbb{Q}_i + \Delta_2 \left[\frac{\mathbb{B}_i^{\text{req}}}{\mathbb{B}_i^{\text{max}}} - \frac{\mathcal{P}_i^{\text{tot}}}{\mathcal{P}_{\text{max}}} \right] \right) \tag{13}$$

where Δ_1 and Δ_2 denote the scaling factors for service offloading. $\mathcal{P}_{\text{max}}$ is the maximum price set by edge service providers. Having computed the net utility for each edge device, the edge device with the maximum net utility value emerges as the winner and get to offload its services first than the others. Thus, without the loss of generality, we can formulate the optimization problem as

$$\text{(P1):} \quad \underset{t>0, \zeta_i(\mathcal{Y},t) \in \{0,1\}}{\text{maximize}} \sum_{i \in N} \mathbb{U}_i \tag{14}$$

$$\begin{aligned} & \mathbb{D}_{to}^t \geq \mathbb{D}_{\text{th}}, i \in N && (15) \\ & \mathbb{B}_i^{\text{req}} \geq \mathbb{B}_i^{th}, i \in N && (16) \\ \text{Subject to} \quad & \mathcal{H}_i(\mathcal{Y}, t) \geq \mathcal{H}_{\text{th}}, i \in N && (17) \\ & \mathcal{Q}_i \geq \mathcal{Q}_{\text{th}}, i \in N && (18) \\ & \mathcal{P}_i^{\text{tot}} \geq \mathcal{P}_{\text{th}}, i \in N. && (19) \end{aligned}$$

Detail description of this approach is discussed. Equation (14) presents the primary optimization function for service offloading. Equation (15) describes that the actual service latency, $\mathbb{D}_{to}^t$, is to be greater than the threshold service latency, $\mathbb{D}_{\text{th}}$. The bandwidth requirement of edge device, $\mathbb{B}_i^{\text{req}}$, is to be greater than the threshold bandwidth requirement, $\mathbb{B}_{\text{th}}$, as shown in (16). Equation (17) represents that the service offloading rate of edge device, $\mathcal{H}_i(\mathcal{Y}, t)$, is to be grater than the threshold offloading rate, $\mathcal{H}_{\text{th}}$. The QoS-level, $\mathcal{Q}_i$, is to be greater than the threshold QoS-level, $\mathcal{Q}_{\text{th}}$, as shown in (19). Equation (17) denotes that the total estimated price for edge device, $\mathcal{P}_i^{\text{tot}}$, is to be grater than the threshold price, $\mathcal{P}_{\text{th}}$. Solving the optimization problem using Lagrangian multipliers, we get

$$\begin{aligned} \Theta_{\mathbb{U}} = & \sum_{i=1}^{N} \frac{\Psi_i}{\mathbb{U}_{\text{th}}} \Gamma_i\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big) \\ & - \xi_1 \left(\sum_{i=1}^{N} \mathbb{B}_i^{\text{req}} - \mathbb{B}_i^{th} \right) - \xi_2 \left(\sum_{i=1}^{N} \mathcal{H}_i(\mathcal{Y}, t) - \mathcal{H}_{\text{th}} \right) \\ & - \xi_3 \left(\sum_{i=1}^{N} \mathcal{Q}_i - \mathcal{Q}_{\text{th}} \right) - \xi_4 \left(\sum_{i=1}^{N} \mathcal{P}_i^{\text{tot}} - \mathcal{P}_{\text{th}} \right) \end{aligned}$$

where ξ_1, ξ_2, ξ_3, and ξ_4 denote the different constraints for Lagrangian multipliers and Ψ_i denotes priority levels of different services in edge devices. Hence, our main objective is to maximize the value of $\mathbb{U}_i$ using the Lagrange multiplier. We have used gradient descent method to solve the problem. We get the Lagrangian Optimization problem is expressed as

$$\begin{aligned} \mathcal{L}_{\Theta_{\mathbb{U}}} = & \sum_{i=1}^{N} \frac{\Psi_i}{\mathbb{U}_{\text{th}}} \mathcal{L}_{\mathbb{U}}\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big) \\ & - \xi_1 \left(\sum_{i=1}^{N} \mathbb{B}_i^{\text{req}} - \mathbb{B}_i^{th} \right) - \xi_2 \left(\sum_{i=1}^{N} \mathcal{H}_i(\mathcal{Y}, t) - \mathcal{H}_{\text{th}} \right) \\ & - \xi_3 \left(\sum_{i=1}^{N} \mathcal{Q}_i - \mathcal{Q}_{\text{th}} \right) - \xi_4 \left(\sum_{i=1}^{N} \mathcal{P}_i^{\text{tot}} - \mathcal{P}_{\text{th}} \right). \end{aligned}$$

Hence, we focus on to optimize $\mathcal{L}_{\mathcal{U}}$ using the Lagrange multiplier. Thus,

$$\frac{\delta \mathcal{L}_{\Theta_{\mathbb{U}}}}{\delta \mathcal{P}_i^{\text{tot}}} = \sum_{i=1}^{N} - \frac{\Psi_i \mathcal{L}_{\mathbb{U}}\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big)}{\mathcal{P}_i^{\text{tot}2}} \tag{20}$$

$$\frac{\delta \mathcal{L}_{\Theta_{\mathbb{U}}}}{\delta \mathcal{H}_i(\mathcal{Y}, t)} = \sum_{i=1}^{N} \frac{\Psi_i}{\mathbb{U}_{\text{th}}} \frac{\delta \mathcal{L}_{\mathbb{U}}\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big)}{\delta \mathcal{H}_i(\mathcal{Y}, t)} \tag{21}$$

$$\frac{\delta \mathcal{L}_{\Theta_{\mathbb{U}}}}{\delta \mathbb{Q}_i} = \sum_{i=1}^{N} \frac{\Psi_i}{\mathbb{U}_{\text{th}}} \frac{\delta \mathcal{L}_{\mathbb{U}}\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big)}{\delta \mathbb{Q}_i} \tag{22}$$

$$\frac{\delta \mathcal{L}_{\Theta_{\mathbb{U}}}}{\delta \mathbb{B}_i^{\text{req}}} = \sum_{i=1}^{N} \frac{\Psi_i}{\mathbb{U}_{\text{th}}} \frac{\delta \mathcal{L}_{\mathbb{U}}\big(\mathcal{H}_i(\mathcal{Y}, t), \mathbb{Q}_i, \mathbb{B}_i^{\text{req}}, \mathcal{P}_i^{\text{tot}}\big)}{\delta \mathbb{B}_i^{\text{req}}}. \tag{23}$$

Algorithm 1 Algorithm for Adaptive Service Offloading

```
Inputs:
  • Set of edge devices ($\mathcal{ED}$), set of services $\mathbb{S}$ and total time $\mathcal{T}$.
Output: Optimized price $\bar{\mathcal{P}}_i^{tot}$ and waiting time $T_{wa}$.
 1: Set $T_{wa} = 0$.
 2: Set $X = \mathcal{ED}$ and $Y = \mathbb{S}$.
 3: for each edge device $ED_i$ do
 4:     if $\mathcal{T} < T_{wa}$ then
 5:         First, approximate the total service latency $\mathbb{D}_{to}^{t}$.
 6:         Create the demand profile $\mathcal{Z}$.
 7:         Estimate average bandwidth requirement $\mathbb{B}_i^{req}$.
 8:         Calculate service offloading rate $\mathcal{H}_i(\mathcal{Y}, t)$.
 9:         Estimate deadline of service $\mathbb{T}_{S_i}$.
10:         Calculate the total data size $\mathcal{X}_i$.
11:         Estimate the total price $\mathcal{P}_i^{tot}$.
12:         Design utility function $\mathbb{U}_i$.
13:         if $\mathbb{U}_i \geq \mathbb{U}_{th}$ then
14:             Updated set of edge devices $\bar{X} = X \cap ED_i$.
15:             Optimized price $\bar{\mathcal{P}}_i^{tot}$ is derived using Eq. 20.
16:             Optimized offloading rate $\mathcal{H}_i(\bar{\mathcal{Y}}, t)$ is derived using Eq. 21.
17:             Optimized QoS-level $\bar{\mathbb{Q}}_i$ is derived using Eq. 22.
18:             Optimized bandwidth $\bar{\mathbb{B}}_i^{req}$ is derived using Eq. 23.
19:             Update waiting time $T_{wa} = T_{wa}$.
20:         else
21:             Updated set of edge devices $\bar{X} = X$.
22:             Non-optimal price cost ($\widehat{\mathcal{P}_i^{tot}}$).
23:             Update waiting time $T_{wa} = T_{wa} + 1$.
24:         end if
25:     end if
26: end for
27: Return $\bar{\mathcal{P}}_i^{tot}$ and $T_{wa}$.
```

Using these equations, we obtain the minimum value of $\Theta_{\mathbb{U}}$ to get the optimal revenue for edge devices. To analyze the overall performance, our proposed service offloading scheme is named as—ADORE for edge devices. Here, we discuss the algorithm for the adaptive service offloading scheme for mobile edge devices. As shown in Algorithm 1, first, we need to provide three inputs: 1) set of edge devices ($\mathcal{ED}$); 2) set of services $\mathbb{S}$; and 3) total time $\mathcal{T}$. In order to provide optimal revenue and price to edge devices, we propose adaptive service offloading scheme to optimize the service latency and price in the network. Initially, we set the waiting time T_{wa} to 0. Thereafter, for each edge device ED_i, we conduct the offloading algorithm. When the total time less than the waiting time, i.e., $\mathcal{T} < T_{\text{wa}}$, then we create a demand profile $\mathcal{Z}$ and also approximate the total service latency $\mathbb{D}_{to}^{t}$. Afterward, we estimate average bandwidth requirement $\mathbb{B}_i^{\text{req}}$ and calculate service offloading rate $\mathcal{H}_i(\mathcal{Y}, t)$. Also, we estimate the deadline of a service $\mathbb{T}_{S_i}$ and calculate the total data size to be offloaded $\mathcal{X}_i$. Further, we estimate the total price $\mathcal{P}_i^{\text{tot}}$. Using the estimated and calculated variables, we design a utility function $\mathbb{U}_i$ for service offloading. If the utility function $\mathbb{U}_i$ greater than the threshold utility function $\mathbb{U}_{\text{th}}$, then we update the set of edge devices $\bar{X} = X \cap \text{ED}_i$. Also, we update the waiting time T_{wa} is updated as well. Along with, we also get the optimized and optimal price cost $\bar{\mathcal{P}}_i^{\text{tot}}$ using Lagrangian multiplier. Similarly, we can get the optimal values for bandwidth, QoS-level and offloading rate using lines 16–18. The process is stopped, when the waiting time crosses a predefined maximum waiting time $T_{\text{wa}}^{\text{max}}$. To optimize the revenue price for edge devices using (14), we use the Lagrangian optimization technique to get the optimal value.

TABLE I
EXPERIMENTAL PARAMETERS

Parameter	Value
Bandwidth	20 MHz
Total number of CPU cycles of computation task	1,000 Megacycles
Service deadline	[4000, 6000] ms
Computation resource demand	[10, 20] MHz
Transmission power of edge device	100 mWatts
Computation capability of edge device	0.7 GHz
Computation capability of the MEC server	100 GHz
Data traffic arrival modeled as Poisson process	[0, 10] unit/sec
Expected size of data traffic	100 Mbits
Computation service arrival (mean size = 1 Mbit)	[0, 10],

Theorem 1: The worst-case computational complexity for ADORE is $O(\mathcal{J}N^2)$, where N is the number of edge devices.

Proof: At first, each edge device tries to offload their computational services to edge servers in order to get the optimal revenues. Therefore, to obtain the optimal revenue, the worst case computation complexity of the service offloading algorithm is $O(\mathbb{X}N^2)$. Before offloading the computational services, the edge devices tries to minimize the offloading latency of network in the absence of multiple edge devices. To minimize the offloading latency, we proposed an service latency approximation algorithm for each edge devices. Hence, the worst case complexity of service latency approximation algorithm is $O(\mathbb{Y}N)$. Thus, combining both the algorithms, we have

$$T(n) = \mathbb{L}_1\{\mathbb{X}T(N^2) + \mathbb{Y}T(N)\} + \mathbb{L}_2 T(1). \tag{24}$$

By combining the worst-case complexity for both the algorithms, we obtain, $O(\mathcal{J}N^2)$, where $\mathcal{J} = \mathbb{X} + \mathbb{Y}$. Hence, we observe that the total computational complexity of ADORE in the worst case, is $O(\mathcal{J}N^2)$ with N as the number of edge devices, which completes the proof of Theorem 1. ■

VI. PERFORMANCE EVALUATION

We present simulation results of the proposed scheme—ADORE[1] in compare to existing schemes. The simulation parameters used in the experiments are shown in Table II.

A. Experimental Setup

1) Parameter Settings: We have listed the experiential setup in Tables I and II. We consider 200 edge devices which are distributed over an area of 1000 m × 1000 m and one macro base station (MBS) co-located to a MEC server. The MEC server located in the MBS, whose computation capability is 100 GHz and the computation capability of edge device is 0.7 GHz. Each base station has 50 orthogonal wireless channels for the computational service offloading from edge devices to edge servers. Here, the cellular backhaul delay coefficient is considered to be 0.0001 sec/kB [31]. The total time duration to offload the computation services of mobile edge devices are randomly distributed between 5 and 10 ms. The corresponding

[1] The adaptive service offloading for revenue maximization scheme for MEC is called ADORE.

TABLE II
EXPERIMENTAL SETTINGS

No.	N	M	$\mathcal{P}_i^{tot}$	$\mathcal{S}_i$	Ψ_i	$\lvert S_{arr}\rvert$
I	200	$10-15$	$[1-300]$	$[5-10]$	$[0,1]$	$[0.2-0.6]$
II	200	$5-10$	$[1-200]$	$[10-15]$	$[0,1]$	$[0.4-0.8]$
III	200	$15-20$	$[1-400]$	$[15-20]$	$[0,1]$	$[0.5-1.0]$

computation file size of each computational service varies within the range 300–800 kB. The delay requirements of edge devices is considered to be within the range 0.5–1 s.

2) Workload: We implement our scheme in ten servers, each machine configured with Intel core-i5 processor and 1.7 GHz CPU. For this paper, we consider two types of traffic workloads: 1) delay-sensitive (i.e., edge services) and 2) delay-tolerant (i.e., cloud services) workloads. Here, the higher priority is given to delay-sensitive traffic than delay-tolerant traffic workloads. Thus, in our setup, we ran edge services at a higher priority than the normal and background services, respectively.

3) Metrics: The service latency in this paper is defined as the total time required to offload the services from edge devices to edge servers. Finally, we design service utilization metric to measure the efficiency of the proposed scheme. The service utilization is defined as the ratio of the total number of services successfully offloaded and total number of services to be offloaded

$$\text{Utilization} = \frac{\text{Total number of services successfully offloaded}}{\text{Total services to be offloaded}}. \tag{25}$$

Larger values indicate the better performance of ADORE, and if the services face the lesser values then it reduces the service utilization. For computational service offloading, we also measure the revenue of the edge devices, and compare ADORE against with other existing schemes.

4) Benchmarks: To evaluate the performance, we use two benchmarks: 1) DESERVE [32] and 2) GREEDY. DESERVE [32] proposed a delay-agnostic service offloading scheme for MEC. They implemented a boosting algorithm using software defined networks, which tries to assign the optimal resources to edge devices and also minimizes the service delay of edge devices. However, they do not consider any resource agnostic property of edge device. We also compared with a GREEDY approach of computational service offloading, here it follows a heuristic solution to find a local optima at each stage with the aim of finding a global optima.

B. Results and Discussion

1) Impact on Total Revenue: Here, we compare the revenue of our proposed scheme—ADORE with two offloading baselines. The figures show that ADORE provides better revenue than the existing schemes—DESERVE and GREEDY under both setting I, II, and III. Fig. 2(a) shows the revenue of the proposed scheme—ADORE for setting I. From the figure, we observe that revenue increases with the increase in the number of edge devices. As the number of edge devices increases then the offloadable services executed by the each edge devices increases, which inherently increases the revenue. However, our proposed scheme—mISO manages to offload the edge services to edge servers form edge devices efficiently, which eventually increases the total revenue. However, ADORE outperforms the other approaches—DESERVE and GREEDY by 5%–9%, respectively. Fig. 2(b) shows the social-welfare of the proposed scheme for setting II. We see that the revenue increases with the increase in the number of edge devices. As the number of services increases, the edge devices and servers get to execute more number of services, hence it inherently increases the revenue. However, it is relatively higher than the existing approaches—DESERVE and GREEDY. Hence, the revenue using the proposed scheme—ADORE is higher than other approaches by 18%–23%. Fig. 2(c) shows the social-welfare of the proposed scheme for setting III. We see that the revenue increases with the increase in the number of edge devices. As the number of edge servers increases, the edge devices get to execute more number of services in a particular time instant, thus the revenue increases. However, we see that the revenue is higher compared to existing approaches—DESERVE and GREEDY. Hence, the revenue using the proposed scheme—ADORE is higher than other approaches by 21%–28%.

2) Impact on Service Latency: Fig. 3(a) shows that the service latency of edge devices, which increases due to increase in real-time IoT applications for setting I. As the edge device increases, therefore the total number of services also increases in the network, which inherently increases the service latency of edge devices. However, the proposed scheme—ADORE outperforms the existing schemes—DESERVE and GREEDY. Hence, the service latency of edge devices using the proposed scheme—ADORE is lesser than other approaches by 4%–6%. Similarly, Fig. 3(b) shows that the service latency of the network with the increase in the number of edge devices using setting II. As the edge devices increases then the congestion in the network also increases, which increases the service latency. However, ADORE outperforms the other approaches—DESERVE and GREEDY by 6%–8%, respectively. Similarly, Fig. 3(c) shows that the service latency of the network with the increase in the number of edge devices using setting III. As the edge services increases, the contention among edge devices also increases to offload their services optimally, which inherently increases the offloading latency. Thus, the total latency increases. However, ADORE outperforms the other approaches—DESERVE and GREEDY by 8%–9%, respectively.

3) Impact on Net Utility: Fig. 4(a) shows the net utility of the proposed scheme—ADORE for setting I. From the figure, we observe that net utility increases with the increase in the number of edge devices. As the number of edge devices increases the service offloading rate and bandwidth requirement also increases. Hence, the net utility also increases, as it a function of both service offloading rate and bandwidth requirement. Therefore, our proposed scheme—ADORE efficiently manages to offload the service to edge servers using this metric. Hence, ADORE outperforms the other approaches—DESERVE and GREEDY by 15%–24%,

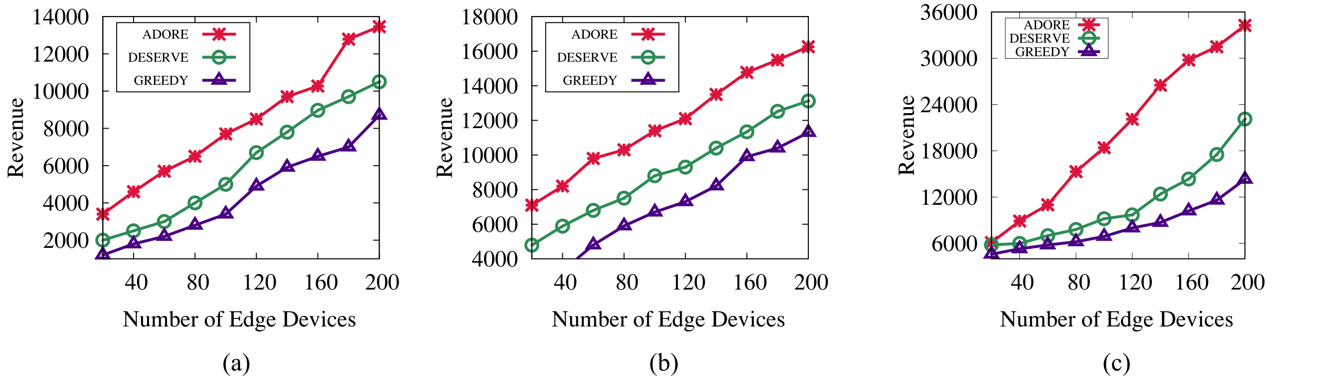


Fig. 2. Analysis of total revenue with settings I, II, and III.

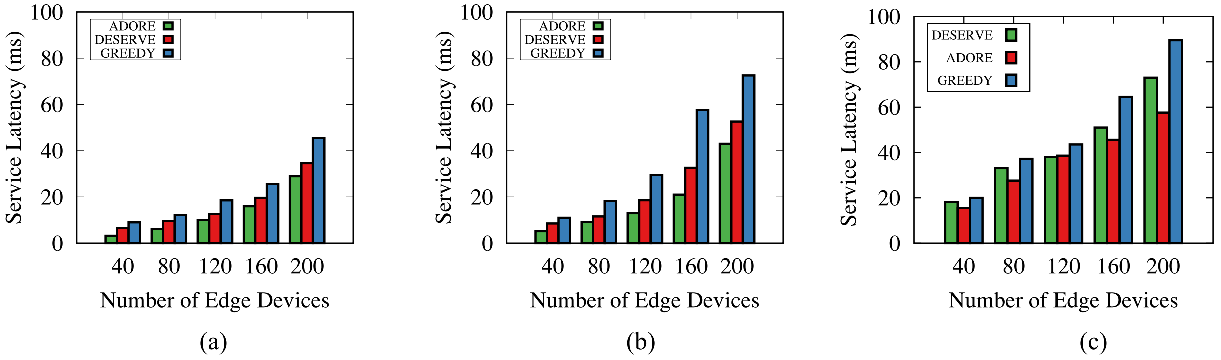


Fig. 3. Analysis of total service latency with settings I, II, and III.

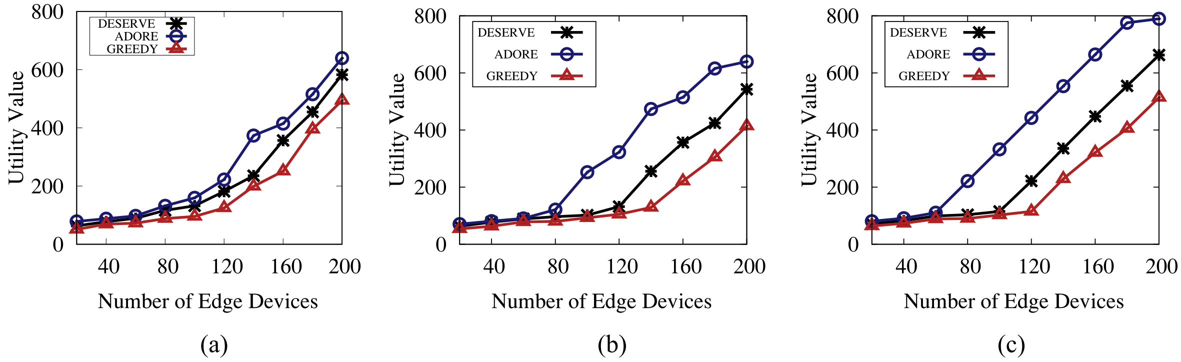
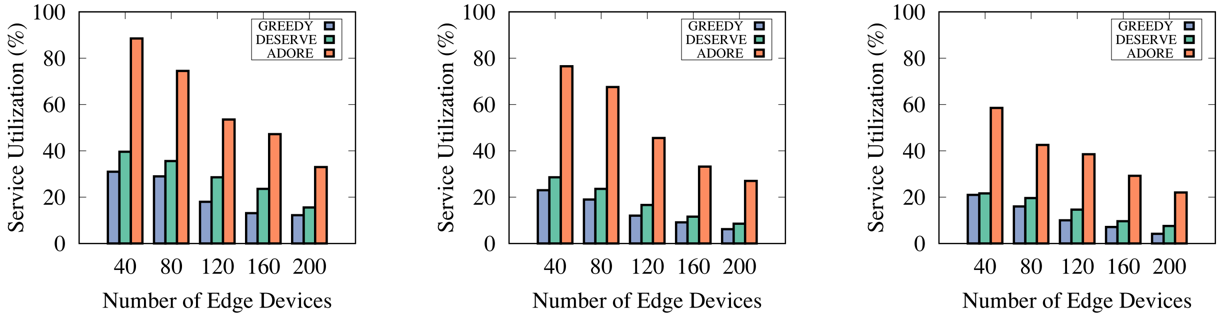


Fig. 4. Analysis of utility value with settings I, II, and III.

respectively. Similarly, we tries to find out the offloading decision for setting II in Fig. 4(b). It also outperforms the existing approaches—DESERVE and GREEDY by 17%–19%, respectively. Similarly, we tries to find out the offloading decision for setting III in Fig. 4(c). ADORE outperforms the existing approaches—DESERVE and GREEDY by 20%–23%, respectively.

4) Impact on Service Utilization: Fig. 5(a) shows the service utilization of the proposed scheme using setting I. We see that the service utilization increases with the increase in the number of edge devices. As the number of edge devices increases, then the designed platform process and executes more number of services at edge servers. Hence, the utilization of the designed platform using the proposed scheme—ADORE is higher than other approaches by 25%–33%. Similarly, Fig. 5(b) shows that the utilization of the proposed scheme—ADORE using setting II. However, the proposed scheme—ADORE outperforms the existing schemes—DESERVE and GREEDY. Hence, the service utilization of designed platform using the proposed scheme—ADORE is higher than other schemes by 28%–38%. Further, Fig. 5(c) shows the utilization of the proposed scheme—ADORE using setting III. As the edge devices and services increases in the network, then the edge devices do not get the enough resources to offload their all services, which inherently decreases the service utilization. However, the service utilization of designed platform using the proposed scheme—ADORE is higher than other schemes by 28%–38%.



(a) (b) (c)

Fig. 5. Analysis of service utilization with settings I, II, and III.

VII. Conclusion

In this paper, we proposed an adaptive service offloading scheme for MEC platform in the presence of multiple edge devices. First, we proposed an optimal service offloading scheme to provide a fair amount of resources to edge devices for efficient service offloading. We also propose a utility maximization scheme to minimize the service latency and price for service offloading. The proposed approach shows remarkable development in terms of net utility, service utilization and revenue. As future work, we will implement the proposed approach with real-bed information and hardware implication. We also propose to have an optimal data dissemination scheme for edge devices in the presence of mobility.

References

[1] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[2] A. Aijaz, "Towards 5G-enabled tactile Internet: Radio resource allocation for haptic communications," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops*, Apr. 2016, pp. 145–150.

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

[4] P. G. Lopez *et al.*, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[5] M. Satyanarayanan *et al.*, "Edge analytics in the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 24–31, Apr./Jun. 2015.

[6] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. Int. Conf. Intell. Syst. Control*, 2016, pp. 1–8.

[7] A. Samanta, S. Bera, and S. Misra, "Link-quality-aware resource allocation with load balance in wireless body area networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 74–81, Mar. 2018.

[8] A. Samanta and S. Misra, "Energy-efficient and distributed network management cost minimization in opportunistic wireless body area networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 376–389, Feb. 2018.

[9] S. Misra and A. Samanta, "Traffic-aware efficient mapping of wireless body area networks to health cloud service providers in critical emergency situations," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2968–2981, Dec. 2018.

[10] A. Samanta and S. Misra, "Dynamic connectivity establishment and cooperative scheduling for QoS-aware wireless body area networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2775–2788, Dec. 2018.

[11] A. Samanta, Y. Li, and S. Chen, "QoS-aware heuristic scheduling with delay-constraint for WBSNs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.

[12] A. Samanta and S. Misra, "EReM: Energy-efficient resource management in body area networks with fault tolerance," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2017, pp. 1–6.

[13] A. Samanta, S. Misra, and M. S. Obaidat, "Wireless body area networks with varying traffic in epidemic medical emergency situation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 6929–6934.

[14] A. Samanta and Y. Li, "Distributed pricing policy for cloud-assisted body-to-body networks with optimal QoS and energy considerations," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2018.2841914.

[15] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[16] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

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

[18] T. Zhao, S. Zhou, X. Guo, and Z. Niu, "Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–7.

[19] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13455–13464, 2017.

[20] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[21] N. T. Ti and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[22] Y. Li, A.-C. Orgerie, I. Rodero, M. Parashar, and J.-M. Menaud, "Leveraging renewable energy in edge clouds for data stream analysis in IoT," in *Proc. IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, 2017, pp. 186–195.

[23] S. Shekhar and A. Gokhale, "Dynamic resource management across cloud-edge resources for performance-sensitive applications," in *Proc. IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, 2017, pp. 707–710.

[24] A. Reiter, B. Prünster, and T. Zefferer, "Hybrid mobile edge computing: Unleashing the full potential of edge computing in mobile device use cases," in *Proc. IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, 2017, pp. 935–944.

[25] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy efficient optimization for computation offloading in fog computing system," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2017, pp. 1–6.

[26] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[27] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[28] A. Samanta and Y. Li, "Time-to-think: Optimal economic considerations in mobile edge computing: Poster," in *Proc. INFOCOM*, 2018, pp. 1–2.

[29] A. Samanta, Z. Chang, and Z. Han, "Latency-oblivious distributed task scheduling for mobile edge computing," in *Proc. IEEE GLOBECOM*, 2018, pp. 1–7.

[30] A. Samanta and Y. Li, "Poster: Latency-oblivious incentive service offloading in mobile edge computing," in *Proc. ACM/IEEE Symp. Edge Comput.*, 2018, p. 3.

[31] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[32] A. Samanta and Y. Li, "DeServE: Delay-agnostic service offloading in mobile edge clouds: Poster," in *Proc. ACM/IEEE Symp. Edge Comput.*, 2017, pp. 1–24.

Amit Samanta (GS'14–M'16) received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, Kolkata, India. He is currently pursuing the M.S. degree at the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India.

He was a Visiting Research Scholar with the Department of Electronic Engineering, Tsinghua University, Beijing, China. He has also been a Junior Project Officer with the Virtual Lab on Advanced Network Technologies (funded by MHRD, Government of India). His current research interests include wireless body area networks, data center networks, cloud computing, and mobile edge computing.

Zheng Chang (S'10–M'13–SM'17) received the B.Eng. degree from Jilin University, Changchun, China, in 2007, the M.Sc. (Tech.) degree from the Helsinki University of Technology (currently, Aalto University), Espoo, Finland, in 2009, and the Ph.D. degree from the University of Jyväskylä, Jyväskylä, Finland, in 2013.

He is currently an Assistant Professor with the University of Jyväskylä. Since 2008, he has held various research positions with the Helsinki University of Technology, University of Jyväskylä, and Magister Solutions Ltd., Jyväskylä. In 2013, he was a Visiting Researcher with Tsinghua University, Beijing, China, for two months. In 2015, he was a Visiting Researcher with the University of Houston, Houston, TX, USA, for one month. His current research interests include Internet of Things, cloud/edge computing, security and privacy, vehicular networks, and green communications.

Dr. Chang was the recipient of awards from the Ulla Tuominen Foundation, the Nokia Foundation, and the Riitta and Jorma J. Takanen Foundation for his excellence in research. He was a recipient of the IEEE Technical Committee on Green Communications and Computing and 23rd Asia–Pacific Conference on Communications in 2017. He serves as an Editor for IEEE ACCESS, *Wireless Networks* (Springer), and IEEE MMTC Communications Frontier, and a Guest Editor for IEEE WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, the IEEE INTERNET OF THINGS JOURNAL, *IEEE Networks*, the *EURASIP Journal on Wireless Communications and Networking*, *Physical Communications and Wireless Communications*, and *Mobile Computing*. He has served as a TPC member for many IEEE major conferences, such as INFOCOM, ICC, and GLOBECOM. He was also named the Exemplary Reviewer of IEEE WIRELESS COMMUNICATIONS LETTER in 2017.