Efficient Computing Resource Sharing for Mobile Edge-Cloud Computing Networks

Yongmin Zhang, *Member, IEEE*, Xiaolong Lan, Ju Ren, *Member, IEEE*, and Lin Cai, *Fellow, IEEE*

***Abstract*— Both the edge and the cloud can provide computing services for mobile devices to enhance their performance. The edge can reduce the conveying delay by providing local computing services while the cloud can support enormous computing requirements. Their cooperation can improve the utilization of computing resources and ensure the QoS, and thus is critical to edge-cloud computing business models. This paper proposes an efficient framework for mobile edge-cloud computing networks, which enables the edge and the cloud to share their computing resources in the form of wholesale and buyback. To optimize the computing resource sharing process, we formulate the computing resource management problems for the edge servers to manage their wholesale and buyback scheme and the cloud to determine the wholesale price and its local computing resources. Then, we solve these problems from two perspectives: i) social welfare maximization and ii) profit maximization for the edge and the cloud. For i), we have proved the concavity of the social welfare and proposed an optimal cloud computing resource management to maximize the social welfare. For ii), since it is difficult to directly prove the convexity of the primal problem, we first proved the concavity of the wholesaled computing resources with respect to the wholesale price and designed an optimal pricing and cloud computing resource management to maximize their profits. Numerical evaluations show that the total profit can be maximized by social welfare maximization while the respective profits can be maximized by the optimal pricing and cloud computing resource management.**

***Index Terms*— Edge, cloud, computing resource sharing, wholesale and buyback, wholesale price.**

I. Introduction

MOBILE applications have changed our lives and become more and more important to our daily living with many new applications appeared and blossomed, e.g., virtual reality, augmented reality, intelligent identification, autonomous driving, interactive gaming, and e-Health [1]–[4]. The rapid increasing computing requirements of mobile applications bring new challenges to the design of mobile devices with limited hardware capabilities. Besides the development of hardware technologies, both the edge and the cloud can provide computing services for mobile devices. Generally, the cloud has tremendous computing resources and thus can handle a large number of computing tasks simultaneously, while the edge can perform qualified computing closer to the source of data and eventually improve the Quality of Service (QoS) by reducing the conveying overhead between users and the cloud. The cooperation of the edge and the cloud can improve the utilization of computing resources and ensure the QoS. Hence, establishing an efficient cooperative edge-cloud network with an appropriate business model is necessary for the future, especially when the development of mobile devices cannot catch up on the growth of the application demands.

Recently, both the Mobile Edge Computing (MEC) and the cloud networks have been extensively studied [5], including system architecture [6]–[9], energy management [10]–[14], data transmission [15]–[17], computing resource optimization [18]–[22], and operation efficiency [23]–[26]. Most of the existing works focused on the design of compatible MEC, reliable cloud networks, and efficient computing task processing protocols, to improve the efficiency of the computing system. However, the cooperation of the MEC and the cloud and the corresponding business model, which are very important for the commercial companies, lacks due attention and may slow down their developments, especially in their starting stage with time-varying and low computing requirements and high investments.

Typically, the MEC has a high average construction and operation cost due to the wide deployment feature. Furthermore, the profitability of the MEC is low since the computing requirements at the MEC typically are time-varying and limited, which leads to a low utilization of computing resources. The cloud has a relatively low average construction and operation cost due to the centralized construction feature and the economies of scale. However, the rapid increasing computing requirements at the cloud not only bring a challenge to the QoS, but also increase the operation cost intensely. In such a case, the cloud wants to obtain computing resources with a low cost while the MEC wants to generate more profit and can provide guaranteed services to the bursty computing tasks. Thanks to the core networks, the MEC and the cloud can be wired connected to share their computing resources with a low communication delay [27], such that they can complement each other to further improve their profitability and QoS.

Our previous work [28] designed an efficient wholesale and buyback scheme (EWBS) for the MEC in mobile edge-cloud computing networks to manage the wholesale and buyback processes. Given the wholesale and buyback prices, the MEC

Manuscript received March 31, 2019; revised September 11, 2019 and January 16, 2020; accepted February 28, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Huang. Date of publication March 25, 2020; date of current version June 18, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFA0706403, in part by the 111 Project under Grant B18059, in part by the National Natural Science Foundation of China (NSFC) under Grant 61702450, Grant 61629302, Grant 61702562, and Grant U19A2067, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by Compute Canada. *(Corresponding author: Ju Ren.)*

Yongmin Zhang and Ju Ren are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China, and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zhangyongmin@csu.edu.cn; renju@csu.edu.cn).

Xiaolong Lan is with the College of Cybersecurity, Sichuan University, Chengdu 610065, China (e-mail: xiaolonglan1112@gmail.com).

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: cai@uvic.ca).

Digital Object Identifier 10.1109/TNET.2020.2979807

servers can make a good trade-off between the wholesale income and the buyback cost to maximize their profitability while satisfying the QoS requirement. However, the effect of the operation cost and the computing requirements at the cloud on the wholesale price has not been considered. In fact, the wholesale price is usually specified by the cloud according to their operation cost and computing requirements. Thus, the wholesale and buyback scheme for the MEC and the pricing scheme for the cloud should be designed jointly.

In the form of wholesale and buyback, the edge servers determine the wholesaled and buyback computing resources according to the wholesale price issued by the cloud and their computing requirements. The cloud adjusts the wholesale price and manages its local computing resources according to the operation cost and the QoS penalty. Considering the case that the edge and the cloud belong to the same entity, we have proved that the social welfare is a concave function of the cloud computing resources and selecting an optimal cloud computing resource can maximize the social welfare. Considering the other case that the edge and the cloud belong to different entities, since it is difficult to directly prove the convexity of the whole problem due to the wholesale, we have proved the concavity of wholesaled computing resources with respect to the wholesale price firstly and then designed an optimal pricing and cloud computing resource management to maximize the profits of the MEC and the cloud. Such that, the profitabilities of the edge and the cloud can be improved by the proposed efficient computing resource sharing schemes. The contributions of our work can be summarized in the following:

- We propose an efficient framework for mobile edge-cloud computing networks, where the MEC and the cloud can share their computing resources with each other in the form of wholesale and buyback.
- We formulate the computing resource management at the MEC and the cloud as profit maximization problems, which are coupled by the wholesale and buyback process.
- We solve the computing resource management problems from two perspectives: i) social welfare maximization by the optimal cloud computing resource management and ii) profit maximization for the MEC and the cloud by the optimal pricing and cloud computing resource management.
- Simulation results show that the proposed computing resource sharing schemes can maximize the total profit by social welfare maximization and the profits of the MEC and the cloud by the optimal pricing and cloud computing resource management.

The rest of the paper is organized as follows: Section II introduces the related works and Section III proposes an efficient framework for mobile edge-cloud computing networks and formulates computing resource management problems for the MEC and the cloud. Section IV considers the computing resource management for the same entity without profit transfers and designs an optimal computing resource management to maximize the social welfare. Section V considers the computing resource management for different entities with profit transfers and proposes an optimal pricing and cloud computing resource management to maximize the respective profits. Section VI demonstrates the efficiency of the proposed algorithm via simulations. Finally, Section VII concludes our work and introduces our future work.

II. Related Works

We introduce the related works from two perspectives: i) system design and ii) system optimization.

System Design

A comprehensive survey of MEC systems, including their architectures and technical enablers, has been presented by [6]. For the network access technology, [8] proposed heuristic link-path formulations, which can design mobile access networks in a reasonable time and [24] proposed an MEC-based object detection architecture via wireless communications for real-time surveillance applications. Taking the scarce energy into consideration, [10] proposed a microwave power transfer based solution for MEC to enable computation in passive low-complexity devices. From the perspective of the construction cost, [25] proposed a heuristic solution to minimize the deployment cost of datacenters under service level objective constraints. Considering the mobility of mobile devices, [7] proposed a novel air-ground integrated mobile edge network (AGMEN) using UAVs to assist the communication, caching, and computing of the edge network, and [17] proposed a two-level edge computing architecture for automated driving services. However, few works consider the combination of computing resources between the MEC and the cloud, which provides an opportunity to substantially enhance the system performance.

System Optimization

Considering the energy limit, [20] proposed a unified design of MEC and wireless power transfer to enhance computation capability and energy supply of mobile devices. Reference [22] proposed an energy-efficient computing offloading management scheme to minimize the energy consumption of mobile devices.

From the perspective of energy consumption, [11] proposed an optimal workload allocation scheme to make a tradeoff between power consumption and transmission delay; [12] proposed an energy-aware offloading scheme to optimize communication and computing resource allocation jointly; [16] proposed an offloading scheme to minimize the overall users' energy consumption under latency constraints. Reference [19] proposed a locally optimal algorithm with the univariate search technique to minimize the energy consumption and the execution latency. [14] and [13] proposed energy-efficient resource allocation schemes for synchronous and asynchronous MECO systems, respectively, to minimize the mobile energy consumption under the constraint on computation latency.

To improve the user experience, [15] designed a joint radio and computational resource allocation scheme to optimize the system performance and improve user satisfaction. Reference [18] proposed a heuristic algorithm to optimize the offloading decision, communication resources, and computing resources jointly. Reference [21] designed truthful, polynomial–time auctions for virtual machines allocation to achieve social welfare maximization and/or the provider's profit maximization. Reference [23] proposed a game-based multi-user computing offloading scheme to maximize multiple users' various interests. Reference [26] designed a family of distributed dynamic cloud network control algorithms, which can jointly schedule computation and communication resources for flow processing and transmission without knowledge of service

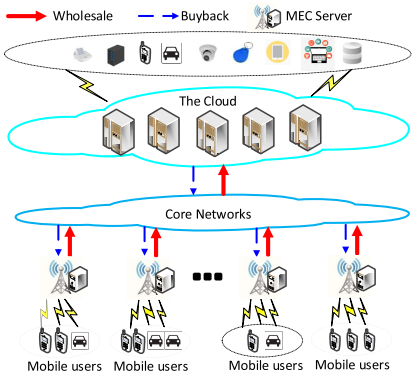


Fig. 1. The architecture for mobile edge-cloud computing networks.

demands, to ensure the stability of the cloud networks and minimize their cost. However, how to jointly optimize the computing resource management of the MEC and the cloud to further improve the utilization of computing resources and their profitability remains an open issue.

III. System Model and Problem Formulation

Considering a mobile edge-cloud computing network, in which there are N_1 MEC servers and N_2 cloud servers. Typically, the MEC servers are built with cellular base stations and provide computing services to the local mobile users while the cloud servers are located around the world and usually provide computing services to the remote users via core networks. Generally, the volume of computing tasks at the cloud is much higher than that at each MEC server, and their latency-sensitivity is typically lower. To guarantee the QoS, the MEC servers need to accomplish the received computing tasks with a tight deadline while the cloud needs to accomplish the received computing tasks as soon as possible.

The MEC servers and the cloud are connected via wired core networks and can share their computing resources with each other based on their computing requirements. Specifically, when the computing resources of the MEC servers are abundant, they can wholesale part of their computing resources to the cloud to improve their profitability. Correspondingly, the cloud can buy computing resources from the MEC servers with a low wholesale price and decrease the cloud computing resources to reduce the high operation cost and improve the QoS. When the computing requirements at the MEC servers are high, they can buy some computing resources back from the cloud with a high buyback price to ensure their QoS.[1] Correspondingly, the cloud needs to satisfy the buyback requests of the MEC servers to increase the profitability. In such a way, the cloud can obtain computing resources from MEC servers by the wholesale scheme with a low wholesale price while the MEC servers can guarantee the delay performance of local bursty traffic and the emergencies by the buyback scheme with a high buyback price. Both the MEC servers and the cloud can be benefited by sharing their computing resources. The architecture for mobile edge-cloud computing networks can be found in Fig. 1.

[1]Paying a higher price can guarantee a high priority of the buyback computing resources [29].

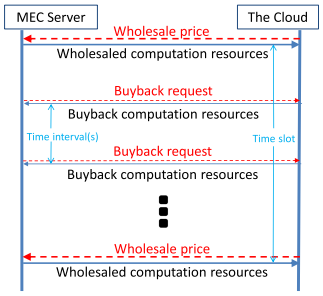


Fig. 2. The operation scheme of mobile edge-cloud computing networks.

A. Operation of Mobile Edge-Cloud Computing Networks

To manage the computing resource sharing processes between the MEC servers and the cloud, we design an efficient framework for mobile edge-cloud computing networks, which is shown in Fig. 2. In this system, there are two different time scales: the first one is time slot and the other is time interval. Typically, one time slot can be tens of minutes while one time interval can be hundreds of milliseconds. At each time slot, the MEC servers can wholesale part of their computing resources to the cloud, which cannot be adjusted during the time slot. At each time interval, the MEC servers can buy some computing resources back from the cloud, which should be satisfied immediately. In this way, the MEC servers need to guarantee the wholesaled computing resources for the cloud and can adjust their buyback computing resources based on their computing requirements. The cloud can utilize the wholesaled computing resources from the MEC during the time slot and need to guarantee the buyback computing resources for the MEC servers during each time interval.

The operation scheme of mobile edge-cloud computing networks will be implemented as follows: At the end of the previous time slot, the cloud needs to issue the wholesale price to the MEC servers. Then, each MEC server determines the wholesaled computing resources and estimates the buyback computing resources. During the time slot, the wholesaled computing resources of the MEC servers are managed by the cloud. If the reserved computing resources at one MEC server cannot satisfy its computing requirements, the MEC server will adjust the buyback computing resources during the coming time intervals by sending a buyback request to the cloud. Then, the cloud will allocate the buyback computing resources to the corresponding MEC server accordingly.

Let t denote the t-th time slot. Divide one time slot into K time intervals and let k denote the k-th time interval during one time slot. Due to the different time granularities of the wholesale and the buyback scheme, the MEC servers need to determine the wholesaled and the buyback computing resources separately [28]. The cloud needs to determine the wholesale and buyback prices and the cloud computing resources at the beginning of each time slot. For simplicity, we assume that the buyback price is given while the wholesale price is adjustable in this paper.

B. Operation Model of MEC Server

The computing resources at each MEC server can be divided into two parts. The first part is reserved by the MEC server and works as a computation server to process the received computing tasks. The second part is wholesaled to the cloud and works as a flexible computation server to process the

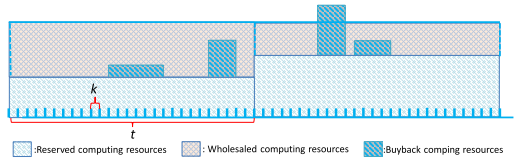


Fig. 3. The computing resource model of the MEC server.

computing tasks from the cloud. Given the randomness of computing tasks, the reserved computing resources may not be sufficient to accomplish all the computing tasks in time, so the MEC server needs to buy some computing resources back from the cloud. Thus, the available computing resources at the MEC server are determined by its wholesale and buyback scheme. The available computing resource at each MEC server is shown in Fig. 3.

Remark: Note that, when the wholesale/buyback event occurs, the computing tasks at the cloud/MEC server will be offloaded to the corresponding MEC servers or/and the cloud for processing. The buyback computing resources for each MEC server can be from itself, other MEC servers, or/and the cloud servers. If the buyback computing resources are from itself, the corresponding amount of wholesaled computing resources will be released to the MEC server, such that all the computing tasks will be processed locally. Otherwise, part of the computing tasks at the MEC server will be offloaded to other MEC servers or/and the cloud for processing. Typically, the cloud will release the wholesaled computing resources back to the MEC servers to reduce the communication overhead. It means that, when an MEC server sends a buyback request to the cloud, the cloud will release the corresponding amount of wholesaled computing resources back to the MEC server if the local computing resources are enough; otherwise, part of computing resources from other MEC servers/the cloud will be allocated to the MEC server.

Generally, the communication delay among the MEC servers and the cloud can be negligible for the following reasons: i) The computing tasks processed locally at the MEC server have no communication delay. ii) Due to the wired connections among the MEC servers and the cloud, the communication delay among them is relatively low compared with that using wireless communications [30], [31]. iii) An offloading computing task may enter a queue of the target server which is busy in serving other computing tasks, so the communication delay can be absorbed by the queueing delay [32]. When the communication delay is significant and cannot be omitted, the MEC server can put the buyback requests forward according to the communication delay to ensure that the overall delay for both computing and communications satisfies the QoS requirement.

Let $C_{e,t}$ denote the total available computing resources, $C^I_{e,t}$ the reserved computing resources, and $C^C_{e,t}$ the wholesaled computing resources, at MEC server e during time slot t, respectively. We have

$$C_{e,t} = C^I_{e,t} + C^C_{e,t}. \tag{1}$$

Let $\hat{C}_{e,k}$ and $\hat{C}^B_{e,k}$ denote the total available computing resources and the buyback computing resources at MEC server e during time interval k, respectively. Thus, we have

$$\hat{C}_{e,k} = \hat{C}^I_{e,k} + \hat{C}^B_{e,k}, \tag{2}$$

where $\hat{C}^I_{e,k} = C^I_{e,t}$ since the reserved computing resources always are available during the entire time slot.

Without loss of generalization, we assume that the arrivals of computing tasks at MEC server e follow a Poisson distribution with an expected value $\lambda_{e,t}$ during time slot t and the computing workload of each computing task follows an exponential distribution with an expected value of R_t. The MEC servers process the computing tasks under the first come first serve (FCFS) policy [33]. Under the FCFS policy, the computation delay of one computing task depends on its arrival time, its computing workload, the prior unprocessed computing workloads, and the available computing resources at the MEC server.

Let $\hat{W}_{e,k}$ denote the computing workloads that are arrived at MEC server e during time interval k and $\hat{Q}_{e,k}$ denote the cumulative unprocessed computing workloads at MEC server e at the end of time interval k, respectively. Thus, we have

$$\hat{Q}_{e,k} = \max(0, \hat{Q}_{e,k-1} + \hat{W}_{e,k} - \hat{C}_{e,k}), \tag{3}$$

Here, the computing workloads are transferred into the requirements of computing resources, i.e., GHz in this paper.

Under the FCFS policy, the computing tasks in $\hat{Q}_{e,k}$ will be accomplished at time interval $k+m$ if $\sum_{k'=k+1}^{k+m-1} \hat{C}_{e,k} < \hat{Q}_{e,k} \leq \sum_{k'=k+1}^{k+m} \hat{C}_{e,k}$, where $\sum_{k'=k+1}^{k+m} \hat{C}_{e,k}$ denotes the total available computing resources for MEC server e during the upcoming time intervals $[k+1, k+m]$. Let $\hat{D}_{e,k}$ denote the maximal computation delay (including the queueing delay) for the computing tasks in $\hat{W}_{e,k}$. For simplicity, the computation delay $\hat{D}_{e,k}$ is defined as

$$\hat{D}_{e,k} = \begin{cases} m, & \text{if } \sum_{k'=k+1}^{k+m-1} \hat{C}_{e,k} < \hat{Q}_{e,k} \leq \sum_{k'=k+1}^{k+m} \hat{C}_{e,k}; \\ 1, & \text{if } \hat{Q}_{e,k} \leq \hat{C}_{e,k+1}, \end{cases} \tag{4}$$

where k' denotes an upcoming time interval at current time interval k. Here, the accuracy of the computation delay $\hat{D}_{e,k}$ depends on the time scale of time interval. To guarantee the QoS at MEC server e, there exists an upper bound on the computation delay $\hat{D}_{e,k}$, denoted by $\bar{D}_{e,t}$. Hence, the following constraint should be satisfied:

$$\bar{D}_{e,t} \geq \hat{D}_{e,k}. \tag{5}$$

Since the available computing resources $\{\hat{C}_{e,k'}, k' \in [k+1, k+m]\}$ determine the computation delay $\hat{D}_{e,k}$, the MEC server can guarantee the QoS by managing the wholesaled and the buyback computing resources.

C. Operation Model of the Cloud

The available computing resources at the cloud include two parts: i) the computing resources at the cloud servers and ii) the computing resources from the MEC servers. In general, the first part depends on the cloud computing resources at the cloud servers, which can be managed by the cloud. The second part depends on the wholesale and buyback scheme of MEC servers, which is affected by the wholesale price.

Let $C_{c,t}$ denote the total available computing resources at the cloud servers during time slot t and $\hat{\mathbf{C}}_k$ denote the total available computing resources at the cloud during time interval k, respectively. The value of $\hat{\mathbf{C}}_k$ can be given by

$$\hat{\mathbf{C}}_k = \hat{C}_{c,k} + \sum_e (\hat{C}^C_{e,k} - \hat{C}^B_{e,k}), \tag{6}$$

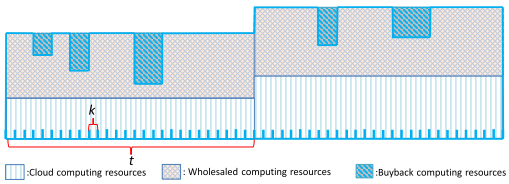


Fig. 4. The computing resource model at the cloud.

where $\hat{C}_{c,k} = C_{c,t}$ and $\hat{C}^C_{e,k} = C^C_{e,t}$ hold since the computing resources at the cloud servers and the wholesaled computing resources from the MEC servers always are available during time slot t. The available computing resource at the cloud is shown in Fig. 4. Generally, there exists an upper bound on the cloud computing resources $C_{c,t}$, denoted by $\bar{C}_{c,t}$. Thus, we have

$$0 \leq C_{c,t} \leq \bar{C}_{c,t}. \tag{7}$$

In this paper, the arrivals of computing tasks at the cloud follow a Poisson distribution with an expected value $\lambda_{c,t}$ during time slot t and the computing workload of each computing task follows an exponential distribution with an expected value of R_t. Considering diverse performance goals of heterogeneous applications at the cloud, priority assignment for various application is one of the popular approaches to improve the efficiency of computing resources [34]. How to assign the priorities of computing tasks is out of the scope of this paper. In this paper, we focus on the average computation delay at the cloud.

Taking the average computation delay as the QoS requirement at the cloud, there exists an upper bound on the average computation delay, denoted by $\bar{D}_{c,t}$. Generally, $\bar{D}_{c,t} > \bar{D}_{e,t}$ always holds. Let $\hat{D}_{c,t}$ denote the average computation delay at the cloud. We have

$$\hat{D}_{c,t} \leq \bar{D}_{c,t}. \tag{8}$$

Here, the computation delay $\hat{D}_{c,t}$ depends on the arrival of computing workloads and the available computing resources $\hat{\mathbf{C}}_k$. Given the arrival of computing workloads, more computing resources $\hat{\mathbf{C}}_k$ lead to lower computation delay $\hat{D}_{c,t}$, which means better QoS.

D. Profit Model of the MEC and the Cloud

For each MEC server, its profit includes four parts: i) the operation cost, ii) the income for processing computing tasks of users; iii) the income for wholesaling computing resources to the cloud; and iv) the cost for buying computing resources back from the cloud. There exist several important factors for the operation cost of the MEC servers and the cloud. We consider these factors, e.g., energy consumption, routine maintenance, and daily expense, as part of the operation cost. For simplicity, we treated the operation cost of each MEC server as a constant depending on the computing resources. Thus, we only consider the last three parts as the profit of MEC servers in this paper.

Let $a_{1,t}$ denote the service fee for processing one unit computing workload during time slot t. Let $U^I_{e,t}$ denote the income for processing computing tasks at MEC server e during time slot t. We have

$$U^I_{e,t} = \sum_k a_{1,t} \hat{W}_{e,k}, \tag{9}$$

since all the computing tasks should be accomplished in time.

Let $a_{2,t}$ denote the wholesale price for wholesaling one unit computing resource to the cloud and $U^S_{e,t}$ denote the total income for wholesaling computing resources during time slot t, respectively. The value of $U^S_{e,t}$ is given by

$$U^S_{e,t} = a_{2,t} K C^C_{e,t}. \tag{10}$$

Let $g(\hat{C}^B_{e,k})$ denote the buyback cost of MEC server e for buying $\hat{C}^B_{e,k}$ unit computing resources back from the cloud during time interval k. To avoid large amount of the buyback computing resources, $g(\hat{C}^B_{e,k})$ usually is an increasing and convex function of $\hat{C}^B_{e,k}$ [35]. Specifically, we set $g(\hat{C}^B_{e,k}) = c_1 \hat{C}^B_{e,k} + c_2 (\hat{C}^B_{e,k})^2$ in this paper. Let $U^B_{e,t}$ denote the total buyback cost of MEC server e during time slot t. We have

$$U^B_{e,t} = \sum_{k=1}^{K} g(\hat{C}^B_{e,k}). \tag{11}$$

Let $\mathbf{U}_{e,t}$ denote the total profit of MEC server e during time slot t. According to the profit model of the MEC, we have

$$\mathbf{U}_{e,t} = U^I_{e,t} + U^S_{e,t} - U^B_{e,t}. \tag{12}$$

It can be found that the total profit $\mathbf{U}_{e,t}$ depends on the wholesaled and buyback scheme of MEC server e.

The profit of the cloud includes four parts: i) the profit for processing computing tasks at the cloud; ii) the local operation cost at the cloud servers; iii) the cost for trading computing resources with the MEC servers, which is given by $\sum_e U^S_{e,t} - U^B_{e,t}$; and iv) the QoS penalty due to the computation delay.

Let $\hat{W}_{c,k}$ denote the computing workloads that are arrived at the cloud during time interval k. We assume that the service fee at the cloud is the same as the MEC. Thus, the profit for processing computing tasks at the cloud, denoted by $U^I_{c,t}$, is $U^I_{c,t} = \sum_k a_{1,t} \hat{W}_{c,k}$.

Let $\hat{g}(C_{c,t})$ denote the local operation cost at the cloud during time slot t. Typically, $\hat{g}(C_{c,t})$ is assumed as an increasing and convex function of $C_{c,t}$. Specifically, we set $\hat{g}(C_{c,t}) = c_3 C_{c,t} + c_4 (C_{c,t})^2$ in this paper. The cloud can determine its local operation cost $\hat{g}(C_{c,t})$ by managing the cloud computing resources $C_{c,t}$ at the cloud servers.

According to the operation model of MEC servers, the cost for trading computing resources with the MEC servers is $\sum_e U^S_{e,t} - U^B_{e,t}$ and the computing resources obtained by the cloud during time interval k is $\sum_e \hat{C}^C_{e,k} - \hat{C}^B_{e,k}$.

At the cloud, computation delay is an important parameter of the QoS. Thus, we define a QoS penalty of computation delay at the cloud, denoted by $U(\hat{D}_{c,t})$, in this paper. Typically, $U(\hat{D}_{c,t})$ is an increasing and convex function of the average computation delay $\hat{D}_{c,t}$.

Thus, the total profit of the cloud, denoted by $\mathbf{U}_{c,t}$, can be given by

$$\mathbf{U}_{c,t} = U^I_{c,t} - \hat{g}(C_{c,t}) - U(\hat{D}_{c,t}) - \sum_e (U^S_{e,t} - U^B_{e,t}). \tag{13}$$

The cloud needs to make a trade-off between the operation costs and the QoS penalty.

E. Problem Formulation

In mobile edge-cloud computing networks, both the MEC servers and the cloud have their operation models and goals. The goal of MEC servers is to maximize their total profit by

trading computing resources with the cloud while guaranteeing the QoS. The goal of the cloud is to maximize the total profit by making a good trade-off between the operation cost and the QoS penalty. Due to their different operation models, their computing resource management problems will be different.

For MEC server e, due to the uncertainty of computing workloads, the MEC server needs to make a trade-off between the wholesale income and the buyback cost. The computing resource management problem for MEC server e can be formulated as

$$\mathbf{P\_1:} \quad \max_{C^C_{e,t},\hat{C}^B_{e,k}} \quad \sum_t \mathbf{U}_{e,t} \tag{14}$$

$$s.t.\ C_{e,t} = C^I_{e,t} + C^C_{e,t}, \quad \forall t, \tag{15}$$

$$\bar{D}_{e,t} \geq \hat{D}_{e,k}, \quad \forall k, t, \tag{16}$$

$$\hat{C}^B_{e,k} \geq 0, \quad \forall k. \tag{17}$$

The objective is to maximize the total profit of the MEC server and the variables are wholesaled and buyback computing resources. The first constraint defines the available range of the wholesaled computing resources. The second constraint ensures that all the computing tasks should be accomplished in time. The third constraint gives the available range of the buyback computing resources at each time interval. Due to the different time granularities of the wholesale and the buyback scheme, the MEC server needs to design the wholesale scheme and the buyback scheme separately.

For the cloud, the computing resources from the MEC servers will be affected by the wholesale price and that from the cloud servers determine the local operation cost. All the available computing resources determine the QoS penalty. To maximize the profit, the cloud needs to make a trade-off among the cloud computing resources, the wholesale price and the QoS penalty. The computing resource management problem for the cloud can be formulated as

$$\mathbf{P\_2:} \quad \max_{a_{2,t},C_{c,t}} \quad \sum_t \mathbf{U}_{c,t} \tag{18}$$

$$s.t.\ 0 \leq C_{c,t} \leq \bar{C}_{c,t}, \quad \forall t, \tag{19}$$

$$\hat{D}_{c,t} \leq \bar{D}_{c,t}, \quad \forall t, \tag{20}$$

$$a_{2,t} \geq 0, \quad \forall t. \tag{21}$$

The objective of the cloud is to maximize the profit and the controllable variables are the wholesale price and the cloud computing resources. The first constraint defines the available range of the cloud computing resources $C_{c,t}$ at cloud servers. The second constraint gives the QoS requirement. The third constraint shows the range of the wholesale price. Generally, the available computing resource depends on the wholesale price $a_{2,t}$ and the cloud computing resources $C_{c,t}$. Thus, the cloud needs to determine the wholesale price $a_{2,t}$ and cloud computing resources $C_{c,t}$ jointly.

From problems **P_1** and **P_2**, it can be found that the computing resource management for the MEC servers and the cloud are associated by the wholesale and buyback scheme via the wholesale price. To maximize their profits, the MEC servers and the cloud should optimize their computing resource managements jointly. In this paper, we solve the above optimization problems from two perspectives: 1) there is no profit transfers between the MEC servers and the cloud, such that the computing resource management problems for the MEC servers and the cloud can be combined and the social welfare will be maximized; 2) the computing resource sharing happens only when the profit transfers occur, such that the MEC and the cloud need to design their computing resource managements based on each others' decision. We introduce the solutions one by one in the following sections.

IV. Wholesale and Buyback Scheme Without Profit Transfers

If all the MEC servers and the cloud belong to the same entity, the profit transfers, e.g., $\sum_e U^B_{e,t} - U^S_{e,t}$, will not affect the total profit of the entity. Thus, the computing resource management problem can be formulated by combining the objectives and the constraints in problems **P_1** and **P_2** together, which can be rewritten as

$$\mathbf{P1:} \quad \min_{C^C_{e,t},\hat{C}^B_{e,k},C_{c,t}} \quad \sum_t \hat{g}(C_{c,t}) + U(\hat{D}_{c,t})$$

$$s.t.\ (15)-(17), (19)-(20), \tag{22}$$

since $\{U^I_{e,t}, \forall e, t\}$ and $\{U^I_{c,t}, \forall t\}$ are constants and total profit transfers $\{\sum_e (U^S_{e,t} - U^B_{e,t}), \forall t\}$ for the entity is zero. The objective of problem **P1** is to minimize the local operation cost and the QoS penalty at the cloud. Furthermore, constraint (21), e.g., $a_{2,t} \geq 0$, can be omitted since there is no profit transfer between the MEC servers and the cloud. To solve this problem, we need to model the relationship between variables and the average computation delay $\hat{D}_{c,t}$ for the computing tasks at the cloud.

For problem **P1**, we have the following lemma:

Lemma 1: The optimal wholesaled computing resources $C^C_{e,t}$ for problem **P1** is $C^C_{e,t} = C_{e,t}$.

This is because the single queue multiple server model is better than the multiple queue multiple server model given the multiplexing gain [36]. It is better for the MEC servers and the cloud to manage the computing resources uniformly. Since all the MEC servers can connect with the cloud via the wired core networks, setting $C^C_{e,t} = C_{e,t}$ can improve the system performance.

It means that the computing resources at both the MEC servers and the cloud servers can be managed by the cloud, such that all the computing tasks will be processed under the management of the cloud. We first model an $M/M/N$ queueing system, in which, $N = N_1 + N_2$, the arrivals of computing tasks follow a Poisson process with $\lambda = \frac{\lambda_{c,t} + \sum_e \lambda_{e,t}}{N}$ during time slot t and the average service rate is $\mu = \frac{C_{c,t} + \sum_e C_{e,t}}{N R_t}$. To ensure the stability of the queueing system, we have the following Lemma:

Lemma 2: The cloud computing resources $C_{c,t}$ should satisfy $C_{c,t} > (\lambda_{c,t} + \sum_e \lambda_{e,t}) R_t - \sum_e C_{e,t}$.

Proof: According to queueing theory, the necessary condition for the queueing system to be stable is that $\rho = \frac{\lambda}{\mu} < 1$. Otherwise, both the queueing length and the computation delay will go infinity. Hence, $C_{c,t} + \sum_e C_{e,t} > (\lambda_{c,t} + \sum_e \lambda_{e,t}) R_t$ should be satisfied. Then, we have $C_{c,t} > (\lambda_{c,t} + \sum_e \lambda_{e,t}) R_t - \sum_e C_{e,t}$ for the cloud computing resources $C_{c,t}$. ■

Given the arrival of computing tasks λ and the service rate μ, the average computation delay $D_{c,t}$ can be given by

$$D_{c,t} = [\frac{(\lambda/\mu)^N \mu}{(N-1)!(N\mu - \lambda)^2}] P_0 + \frac{1}{\mu}, \tag{23}$$

where

$$P_0 = [\sum_{n=0}^{N-1} \frac{(N\lambda)^n}{n!\mu^n} + \frac{(N\lambda)^N}{N!\mu^N}\frac{\mu}{\mu-\lambda}]^{-1}. \quad (24)$$

Here, $D_{c,t}$ is the expected computation delay for the computing tasks at both the MEC servers and the cloud servers.

Note that, the computing tasks at the MEC servers and the cloud have different QoS requirements. For the computing tasks at the MEC servers, all of them should be accomplished before their deadlines. For the computing tasks at the cloud, they should be accomplished as soon as possible. Given the set of computing tasks and their distributions of computing workloads, the average computation delay for both the MEC servers and the cloud servers is given by $D_{c,t}$. However, it is difficult to obtain the average computation delay $\hat{D}_{c,t}$ for the computing tasks at the cloud servers directly. Thus, we make the following task scheduling and approximation:

Task Scheduling

When the computation delays (including the queueing delays) of all the computing tasks at the MEC servers are within their maximum tolerable delay bounds, all the computing tasks will be processed under the FCFS policy. Otherwise, it will be processed with a high priority at other MEC servers/the cloud to ensure that all the computing tasks at the MEC servers can be accomplished in time. Since the cloud is assumed to have sufficient resources to handle all requests, no computing task will be blocked. Hence, the average computation delay for the computing tasks at MEC server e can be estimated by $\min\{D_{c,t}, \bar{D}_{e,t}\}$. According to Little's law [37], the long-term average number of computing tasks in a stationary system is equal to the long-term average effective arrival rate $\lambda_{c,t}+\sum_e \lambda_{e,t}$ multiplied by the average time $D_{c,t}$ that a computing task spends in the system. As there is no blocked task in the queueing system, the average computation delay $\hat{D}_{c,t}$ for the computing tasks those arrival at the cloud can be estimated by

$$\hat{D}_{c,t} \approx \frac{D_{c,t}(\lambda_{c,t}+\sum_e \lambda_{e,t}) - \sum_e \lambda_{e,t}\min\{D_{c,t}, \bar{D}_{e,t}\}}{\lambda_{c,t}}. \quad (25)$$

It can be found that $\hat{D}_{c,t}$ can be treated as a linear function of the average computation delay $D_{c,t}$.

Let $\hat{Q}_k$ denote the cumulative unprocessed computing workloads from both the MEC and the cloud in the M/M/N queueing system. Let k' denote an upcoming time interval at current time interval k. We can find a k' that satisfies

$$\Big(k'-1\Big)\Big(C_{c,t}+\sum_e C_{e,t}\Big) \le \hat{Q}_k \le k'\Big(C_{c,t}+\sum_e C_{e,t}\Big). \quad (26)$$

To ensure that all the computing tasks at the MEC can be accomplished before their deadlines, the buyback computing resources $\hat{C}^B_{e,k}$ can be set by

$$\begin{cases} \hat{C}^B_{e,k+\bar{D}_{e,t}} = \hat{W}_{e,k}, & \text{if } k' \ge \bar{D}_{e,t}; \\ \hat{C}^B_{e,k'} = \hat{W}_{e,k}, & \text{otherwise.} \end{cases} \quad (27)$$

when $\hat{W}_{e,k} \le \hat{C}^C_{e,k}$. Otherwise, computing resources $\hat{W}_{e,k} - \hat{C}^C_{e,k}$ will be bought from other MEC servers/the cloud during time interval $k+\bar{D}_{e,t}-\delta$, where δ is the communication delay for the computing task offloading. It means that the computing tasks at the MEC servers will be given a higher priority only when they cannot be accomplished in time under the FCFS policy.

By now, the optimal $C^C_{e,t}$ and $\hat{C}^B_{e,k}$ have been obtained and the problem **P1** can be rewritten as

$$\textbf{P1':} \min_{C_{c,t}} \sum_t \hat{g}(C_{c,t}) + U(\hat{D}_{c,t}) \quad (28)$$

$$s.t.\ \ 0 \le C_{c,t} \le \bar{C}_{c,t}, \quad \forall t, \quad (29)$$

$$\hat{D}_{c,t} \le \bar{D}_{c,t}, \quad \forall t, \quad (30)$$

The objective is to minimize the local operation cost and the QoS penalty at the cloud while the controllable variable is the cloud computing resources $C_{c,t}$. Since $\hat{g}(C_{c,t})$ is an increasing and convex function of $C_{c,t}$, we needs to analyze the convexity of $U(\hat{D}_{c,t})$ with respect to $C_{c,t}$.

First, we analyze the convexity of the average computation delay $D_{c,t}$ with respect to the cloud computing resources $C_{c,t}$ and have the following lemma:

Lemma 3: The average computation delay $D_{c,t}$ is a deceasing and convex function of the cloud computing resources $C_{c,t}$.

Proof: According to the definition of Erlang's C formula, the probability, denoted by $C(N, \lambda/\mu)$, that an arriving computing task should wait in the queue since all the servers are busy, can be given by

$$C(N,\lambda/\mu) = \frac{(\lambda/\mu)^N \mu}{(N-1)!(N\mu-\lambda)} P_0, \quad (31)$$

which has proved to be an increasing and convex function of λ/μ [38]–[40]. Thus, $\frac{\partial C(N,\lambda/\mu)}{\partial(\lambda/\mu)} > 0$ and $\frac{\partial^2\ C(N,\lambda/\mu)}{\partial(\lambda/\mu)^2} > 0$ hold. We can derive that $\frac{\partial C(N,\lambda/\mu)}{\partial \mu} < 0$ and $\frac{\partial^2\ C(N,\lambda/\mu)}{\partial \mu^2} > 0$. According to (23), $D_{c,t}$ can be rewritten as

$$D_{c,t} = \frac{C(N,\lambda/\mu)}{N\mu-\lambda} + \frac{1}{\mu}. \quad (32)$$

Thus, we have

$$\frac{\partial D_{c,t}}{\partial \mu} = \frac{\frac{\partial C(N,\lambda/\mu)}{\partial \mu}(N\mu-\lambda) - NC(N,\lambda/\mu)}{(N\mu-\lambda)^2} - \frac{1}{\mu^2}. \quad (33)$$

and

$$\frac{\partial^2\ D_{c,t}}{\partial \mu^2} = \frac{\frac{\partial^2\ C(N,\lambda/\mu)}{\partial \mu^2}}{N\mu-\lambda} - \frac{2N\frac{\partial C(N,\lambda/\mu)}{\partial \mu}}{(N\mu-\lambda)^2} + \frac{NC(N,\lambda/\mu)}{(N\mu-\lambda)^3} + \frac{2}{\mu^3}.$$

It can be found that $\frac{\partial D_{c,t}}{\partial \mu} < 0$ and $\frac{\partial^2\ D_{c,t}}{\partial \mu^2} > 0$. Thus, $D_{c,t}$ is a decreasing and convex function of μ. Since μ is an increasing and linear function of $C_{c,t}$, the average computation delay $D_{c,t}$ is a deceasing and convex function of the cloud computing resources $C_{c,t}$. ■

According to (25), since $\hat{D}_{c,t}$ can be treated as a linear function of $D_{c,t}$, $\hat{D}_{c,t}$ is a decreasing and convex function of $C_{c,t}$. Since $U(\hat{D}_{c,t})$ is an increasing and convex function of $\hat{D}_{c,t}$, $U(\hat{D}_{c,t})$ is a decreasing and convex function of $C_{c,t}$. Since both the objective function and the constraints are convex with respect to $C_{c,t}$, problem **P1'** is a convex optimization problem, which can be solved by the existing tools, e.g., Fmincon in Matlab.

By now, the optimal $C^C_{e,t}$, $\hat{C}^B_{e,k}$ and $C_{c,t}$ have been obtained: $C^C_{e,t}$ is $C_{e,t}$, $\hat{C}^B_{e,k}$ is obtained by (27), and the optimal $C_{c,t}$

Algorithm 1 Optimal Cloud Computing Resource Management (Social Welfare Maximization)

Input: $(\lambda_{e,t}, R_t, \bar{D}_{e,t})$ for computing tasks, $(C_{e,t}, K, T)$ for each MEC server, and $(\bar{C}_{c,t}, \bar{D}_{c,t})$ for the cloud;
Output: $\{C_{c,t}, C^C_{e,t} \forall t\}$ and $\{\hat{C}^B_{e,k} \forall k\}$;
for *each time slot* t **do**
  1) Set $C^C_{e,t} = C_{e,t}$;
  2) Calculate optimal $C_{c,t}$ by solving problem **P1'** using Fmincon;
  for *each time interval* k **do**
    Set $\hat{C}^B_{e,k}$ by Eq. (27);
  end
end

is obtained by solving problem **P1'**, respectively. Such that, the social welfare can be maximized by the optimal cloud computing management, sketched as Algorithm 1. However, in this scenario, the entity needs to build both the MEC and the cloud, which is costly and even infeasible. Thus, we study the wholesale and buyback scheme with profit transfers.

V. Wholesale and Buyback Scheme With Profit Transfers

Generally, the MEC and the cloud belong to different entities or different departments in a entity with their own profit objectives. To achieve their respective objectives, the MEC and the cloud can share their computing resources with profit transfers. In this paper, the MEC intends to maximize the profit by providing computing services to mobile users and wholesaling their abundant computing resources to the cloud, while the cloud intends to provide better computing services to their customers and reduce the operation cost by buying computing resources from the MEC servers and managing the cloud computing resources. From problems **P_1** and **P_2**, it can be found that the MEC servers determine their wholesale and buyback scheme while the cloud determines the wholesale price and the cloud computing resources. These two problems are coupled by profit transfers under the wholesaled and buyback scheme. In addition, due to profit transfers, it is difficult to prove the convexity of these two problems.

According to the operation scheme of mobile edge-cloud computing networks in Fig. 2, the cloud needs to issue a wholesale price to the MEC servers, and then the MEC servers determine their wholesaled computing resources. To solve these problems, we first analyze the relationship between the wholesale price and the wholesaled computing resources from the MEC servers. Then, we drive a necessary condition for the optimal available computing resources. Finally, we design an optimal pricing and the cloud computing resource management to maximize the profits of the MEC and the cloud simultaneously.

Note that, in this section, the expected computation delays for all the computing tasks can be calculated by two kinds of queueing systems. The first one is for the computing tasks that are processed by the reserved computing resources at each MEC server, which can be modeled as an M/M/1 queueing system. Another one is for the computing tasks uploaded by the MEC servers and those arrival at the cloud, which can be modeled as an M/M/N queueing system in Section IV.

A. Relationship Between Wholesale Price and Wholesaled Computing Resources

Given the arrival of computing tasks, the wholesaled and the buyback computing resources are coupled by the computation delay $\hat{D}_{e,k}$. Given the reserved computing resources $C^I_{e,t}$ and the deadline $\bar{D}_{e,t}$, the expected buyback computing resources during time slot t, denoted by $\bar{C}^B_{e,t}$, can be obtained by

$$\bar{C}^B_{e,t} = K \frac{\lambda_{e,t} R_t^2}{(C^I_{e,t} - \lambda_{e,t} R_t)\bar{D}_{e,t}} e^{(\lambda_{e,t} - \frac{C^I_{e,t}}{R_t})\bar{D}_{e,t}}, \quad (34)$$

and the minimal expected buyback cost $U^B_{e,t}$, denoted by $\bar{U}^B_{e,t}$, can be given by (35), as shown at the bottom of the page, [28]. Both of $\bar{C}^B_{e,t}$ and $\bar{U}^B_{e,t}$ are decreasing and convex function of the reserved computing resources $C^I_{e,t}$ [28]. Since $C^C_{e,t} = C_{e,t} - C^I_{e,t}$, $\bar{U}^B_{e,t}$ is an increasing and convex function of the wholesaled computing resources $C^C_{e,t}$.

Based on the relationship between $\bar{U}^B_{e,t}$ and $C^C_{e,t}$, we analyze the relationship between the wholesaled computing resources $C^C_{e,t}$ and the wholesale price $a_{2,t}$, and have the following theorem:

Theorem 1: Given the arrival of computing tasks, the wholesaled computing resources $C^C_{e,t}$ is a non-decreasing function of the wholesale price $a_{2,t}$.

Proof: According to the profit model, $\mathbf{U}_{e,t}$ includes three parts: $U^I_{e,t}$, $U^S_{e,t}$, and $-U^B_{e,t}$, given by (9)-(11), respectively. $U^I_{e,t}$ can be treated as a constant since the arrival of computing tasks and their expected workloads are given. Given the value of $a_{2,t}$, $U^S_{e,t}$ is a linear and increasing function of $C^C_{e,t}$ since $U^S_{e,t} = a_{2,t} K C^C_{e,t}$. $-U^B_{e,t}$ is a decreasing and concave function of $C^C_{e,t}$. Thus, according to the results in [28], the total profit $\mathbf{U}_{e,t}$ is a concave function of $C^C_{e,t}$ and the optimal $C^C_{e,t}$ satisfies

$$\begin{cases} C^C_{e,t} = C_{e,t} - \lambda_{e,t} R_t, & \text{if } \frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}}|_{C^I_{e,t} = \lambda_{e,t} R_t} > 0; \\ C^C_{e,t} = 0, & \text{if } \frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}}|_{C^I_{e,t} = C_{e,t}} < 0; \\ \frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}} = 0, & \text{otherwise.} \end{cases} \quad (36)$$

With the increase of $a_{2,t}$, $\frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}}$ increases and the corresponding $C^C_{e,t}$ is non-decreasing for the following reason: For the first case in (36), $C^C_{e,t}$ is a constant since it has reached its maximal value; For the second case in (36), $C^C_{e,t}$ will increase when $\frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}}|_{C^I_{e,t} = C_{e,t}} > 0$; For the third case in (36), $C^C_{e,t}$ will increase to make sure that $\frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}} = 0$. Thus, the wholesaled

$$\bar{U}^B_{e,t} = K e^{(\lambda_{e,t} - \frac{C^I_{e,t}}{R_t})\bar{D}_{e,t}} \Big(\frac{c_1 \lambda_{e,t} R_t^2}{(C^I_{e,t} - \lambda_{e,t} R_t)\bar{D}_{e,t}} + \frac{2 c_2 \lambda_{e,t} R_t^3 C^I_{e,t}}{\big((C^I_{e,t} - \lambda_{e,t} R_t)\bar{D}_{e,t}\big)^2} \Big) \quad (35)$$

computing resources $C^C_{e,t}$ is non-decreasing with the increase of the wholesale price $a_{2,t}$. ■

It means that, when the cloud increases the wholesale price $a_{2,t}$, the wholesaled computing resources from the MEC servers will not decrease. Based on Theorem 1, we have the following lemma:

Lemma 4: Given the arrival of computing tasks, the wholesaled computing resources $C^C_{e,t}$ is an increasing and concave function of the wholesale price $a_{2,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$.

Proof: According to (36), $\frac{\partial \mathbf{U}_{e,t}}{\partial C^C_{e,t}} = 0$ is the sufficient condition for optimal $C^C_{e,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$. For the components of total profit $\mathbf{U}_{e,t}$, we have $\frac{\partial U^I_{e,t}}{\partial C^C_{e,t}} = 0$, $\frac{\partial U^S_{e,t}}{\partial C^C_{e,t}} = a_{2,t}$, $\frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}} > 0$, and $\frac{\partial^2 \bar{U}^B_{e,t}}{\partial (C^C_{e,t})^2} > 0$. For the optimal $C^C_{e,t}$, we have $a_{2,t} = \frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}}$. Since $\frac{\partial^2 \bar{U}^B_{e,t}}{\partial (C^C_{e,t})^2} > 0$, $\frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}}$ is an increasing function of $C^C_{e,t}$. It means that, with the increase of $C^C_{e,t}$, $\frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}}$ will be much higher. Thus, $a_{2,t}$ is an increasing function of $C^C_{e,t}$. Let $C^C_{e,t}(a_{2,t})$, $C^{C'}_{e,t}(a'_{2,t})$ and $C^{C''}_{e,t}(a''_{2,t})$ denote the optimal wholesaled computing resources with the wholesale prices $a_{2,t}$, $a'_{2,t}$ and $a''_{2,t} = 2a'_{2,t} - a_{2,t}$, respectively. We have $C^{C'}_{e,t}(a'_{2,t}) > \frac{1}{2}(C^C_{e,t}(a_{2,t}) + C^{C''}_{e,t}(a''_{2,t}))$. Thus, $C^C_{e,t}$ is an increasing and concave function of $a_{2,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$. ■

Given the wholesale and buyback prices and the arrival of computing tasks, the optimal reserved and wholesaled computing resources $C^I_{e,t}$ and $C^C_{e,t}$ can be obtained by (36) and the expected buyback computing resources $\bar{C}^B_{e,t}$ can be obtained by (34). Note that, for the third case in (36), problem **P_1** is a convex optimization problem and can be solve by the existing tools, such as Bisection method in [28] and sub-gradient methods in [41].

B. Minimal Operation Cost for the Cloud

Besides the wholesaled computing resources from the MEC servers, the cloud can generate computing resources by their own servers. The total computing resources at the cloud $\hat{\mathbf{C}}_k$ during time interval k is given by (6), in which $\hat{C}_{c,k}$ is determined by the local operation cost $\hat{g}(C_{c,t})$ while $\sum_e(\hat{C}^C_{e,k} - \hat{C}^B_{e,k})$ are determined by the wholesale price $a_{2,t}$. Note that, the expected $\sum_k \hat{C}^B_{e,k}$ is given by $\bar{C}^B_{e,t}$ in (34).

For the cloud, the available computing resources from MEC server e during time slot t is $C^C_{e,t} - \bar{C}^B_{e,t}$. For the value of $C^C_{e,t} - \bar{C}^B_{e,t}$, we have the following lemma:

Lemma 5: The available computing resources $C^C_{e,t} - \bar{C}^B_{e,t}$ is an increasing and concave function of $C^C_{e,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$.

Proof: It can be proved that the buyback computing resources $\bar{C}^B_{e,t}$ is an increasing and convex function of $C^C_{e,t}$. Comparing (34) and (35), it can be found that $\bar{U}^B_{e,t} = (c_1 + \frac{c_2 R_t C^I_{e,t}}{(C^I_{e,t} - \lambda_{e,t}R_t)D_{e,t}})\bar{C}^B_{e,t}$. According to the proof of Theorem 1, $a_{2,t} = \frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}}$ is a sufficient condition for optimal $C^C_{e,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$. Since $\frac{\partial \bar{U}^B_{e,t}}{\partial C^C_{e,t}} > c_1 \frac{\partial \bar{C}^B_{e,t}}{\partial C^C_{e,t}}$ and $c_1 > a_{2,t}$, $\frac{\partial \bar{C}^B_{e,t}}{\partial \hat{C}^C_{e,t}} < 1$ always holds. Thus, $C^C_{e,t} - \bar{C}^B_{e,t}$ is an increasing and concave function of $C^C_{e,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$. ■

Given the wholesale price $a_{2,t}$, the expected computing resources from MEC servers can be given by $\sum_e(C^C_{e,t} - \bar{C}^B_{e,t})$. For the relationship between the available computing resources $C^C_{e,t} - \bar{C}^B_{e,t}$ and the wholesale price $a_{2,t}$, we have

Lemma 6: The available computing resources $C^C_{e,t} - \bar{C}^B_{e,t}$ is an increasing and concave function of the wholesale price $a_{2,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$.

Proof: According to Lemma 4, when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$, the wholesaled computing resources $C^C_{e,t}$ is an increasing and concave function of the wholesale price $a_{2,t}$. Thus, we have $\frac{\partial C^C_{e,t}}{\partial a_{2,t}} > 0$ and $\frac{\partial^2 C^C_{e,t}}{\partial a^2_{2,t}} < 0$. According to Lemma 5, the available computing resources $C^C_{e,t} - \bar{C}^B_{e,t}$ is an increasing and concave function of the wholesaled computing resources $C^C_{e,t}$. We have $\frac{\partial C^C_{e,t} - \bar{C}^B_{e,t}}{\partial C^C_{e,t}} > 0$ and $\frac{\partial^2 C^C_{e,t} - \bar{C}^B_{e,t}}{\partial (C^C_{e,t})^2} < 0$. Then, we can derive that $\frac{\partial C^C_{e,t} - \bar{C}^B_{e,t}}{\partial a_{2,t}} = \frac{\partial C^C_{e,t} - \bar{C}^B_{e,t}}{\partial C^C_{e,t}} \frac{\partial C^C_{e,t}}{\partial a_{2,t}} > 0$ and $\frac{\partial^2 C^C_{e,t} - \bar{C}^B_{e,t}}{\partial a^2_{2,t}} = \frac{\partial^2 C^C_{e,t} - \bar{C}^B_{e,t}}{\partial (C^C_{e,t})^2}(\frac{\partial C^C_{e,t}}{\partial a_{2,t}})^2 + \frac{\partial C^C_{e,t} - \bar{C}^B_{e,t}}{\partial (C^C_{e,t})} \frac{\partial^2 C^C_{e,t}}{\partial a^2_{2,t}} < 0$. Thus, the available computing resources $C^C_{e,t} - \bar{C}^B_{e,t}$ is an increasing and concave function of the wholesale price $a_{2,t}$ when $C^C_{e,t} \in (0, C_{e,t} - \lambda_{e,t}R_t)$. ■

It means that, with the increase of the wholesale price $a_{2,t}$, the available computing resource $\sum_e C^C_{e,t} - \bar{C}^B_{e,t}$ is non-decreasing and concave.

The cost for the cloud to obtain the computing resources $\sum_e C^C_{e,t} - \bar{C}^B_{e,t}$ from MEC servers is $\sum_e a_{2,t} C^C_{e,t} - \bar{U}^B_{e,t}$ and the local operation cost for the cloud servers to generate $C_{c,t}$ is $\hat{g}(C_{c,t})$. Let $p_{e,t}(a_{2,t})$ and $p_{c,t}(C_{c,t})$ denote the minimal cost for obtaining more computing resources from the MEC and the cloud servers, respectively. Then, we have

$$p_{e,t}(a_{2,t}) = \frac{\sum_e (U^{S'}_{e,t} - \bar{U}^{B'}_{e,t}) - \sum_e (U^S_{e,t} - \bar{U}^B_{e,t})}{\sum_e (C^{C'}_{e,t} - \bar{C}^{B'}_{e,t}) - \sum_e (C^C_{e,t} - \bar{C}^B_{e,t})}, \quad (37)$$

$$p_{c,t}(C_{c,t}) = \frac{\partial \hat{g}(C_{c,t})}{\partial C_{c,t}}, \quad (38)$$

where $\{U^{S'}_{e,t}, \bar{U}^{B'}_{e,t}, C^{C'}_{e,t}, \bar{C}^{B'}_{e,t}\}$ are the parameters with the wholesale price $a'_{2,t}$ satisfying $a'_{2,t} > a_{2,t}$, and $\sum_e(C^{C'}_{e,t} - \bar{C}^{B'}_{e,t}) - \sum_e(C^C_{e,t} - \bar{C}^B_{e,t})$ is a small enough constant.

To minimize the total operation cost, we have the following theorem:

Theorem 2: There exists an Equilibrium point satisfying $p_{e,t}(a_{2,t}) = p_{c,t}(C_{c,t})$ when $C_{c,t} < \bar{C}_{c,t}$.

Proof: Defined ϵ as a small enough constant. To obtain ϵ unit computing resources from MEC servers, the wholesale price should be increased from $a_{2,t}$ to $a'_{2,t}$ and the cost for buying ϵ unit computing resources is $p_{e,t}(a_{2,t})$. To generate ϵ unit computing resources by the cloud servers, the local operation cost can be treated as $p_{c,t}(C_{c,t})$. Given the total available computing resources $\hat{\mathbf{C}}_k$, $p_{e,t}(a_{2,t}) = p_{c,t}(C_{c,t})$ always holds when $C_{c,t} < \bar{C}_{c,t}$ for the following reason: 1) If $p_{c,t}(C_{c,t}) > p_{e,t}(a_{2,t})$, it means that there exists a higher wholesale price $a_{2,t}$ for the cloud to buy more computing resources from MEC servers to reduce the total operation cost. 2) If $p_{c,t}(C_{c,t}) < p_{e,t}(a_{2,t})$, it means that the cloud

can generate more cloud computing resources to reduce the total operation cost while reducing the wholesale price $a_{2,t}$. Thus, there exists an Equilibrium point satisfying $p_{e,t}(a_{2,t}) = p_{c,t}(C_{c,t})$ when $C_{c,t} < \bar{C}_{c,t}$. ■

According to Theorem 2, with the increase of the total available computing resources $\hat{\mathbf{C}}_k$, the costs $p_{e,t}(a_{2,t})$ and $p_{c,t}(C_{c,t})$ will increase simultaneously until $C_{c,t} = \bar{C}_{c,t}$. However, when the cloud computing resources $C_{c,t}$ reaches its upper bound $\bar{C}_{c,t}$, the cloud only can obtain the computing resources from the MEC by increasing the wholesale price $a_{2,t}$. Thus, we have the following lemma:

Lemma 7: The operation solution satisfies $p_{e,t}(a_{2,t}) \geq p_{c,t}(C_{c,t})$ when $C_{c,t} = \bar{C}_{c,t}$.

Proof: To minimize the total operation cost, according to Theorem 2, the cloud will adjust $a_{2,t}$ and the $C_{c,t}$ to make sure $p_{e,t}(a_{2,t}) = p_{c,t}(C_{c,t})$ until the cloud computing resources $C_{c,t}$ reaches its upper bound $\bar{C}_{c,t}$. If the QoS penalty $U(\hat{D}_{c,t})$ is very high or the constraint $\hat{D}_{c,t} \leq \bar{D}_{c,t}$ cannot be satisfied, the cloud has to buy more computing resources from MEC servers by increasing the wholesale price $a_{2,t}$. Since $p_{e,t}(a_{2,t})$ is an increasing function of $a_{2,t}$, we have $p_{e,t}(a_{2,t}) \geq p_{c,t}(C_{c,t})$ when $C_{c,t} = \bar{C}_{c,t}$. ■

C. Optimal Pricing and Cloud Computing Resource Management

Given the wholesale price $a_{2,t}$, the value of $\sum_e (C^C_{e,t} - \bar{C}^B_{e,t})$ is determined. Thus, given any wholesale price $a_{2,t}$, we can calculate the optimal cloud computing resources $C_{c,t}$ by solving the following problem:

$$\begin{aligned} \textbf{P2\_1:} \min_{C_{c,t}} \quad & \hat{g}(C_{c,t}) + U(\hat{D}_{c,t}) \\ s.t. \quad & 0 \leq C_{c,t} \leq \bar{C}_{c,t}, \quad \forall t, \\ & \hat{D}_{c,t} \leq \bar{D}_{c,t}, \quad \forall t. \end{aligned}$$

The objective function is to minimize the total local operation cost and the QoS penalty at the cloud while the constraints defining the ranges for the cloud computing resources $C_{c,t}$ and the computation delay $\hat{D}_{c,t}$.

Given the wholesale price $a_{2,t}$ and the cloud computing resources $C_{c,t}$, the expected computation delay $\hat{D}_{c,t}$ can be calculated by (23) via setting $\lambda = \lambda_{c,t}$ and $\mu = \frac{C_{c,t} + \sum_e (C^C_{e,t} - \bar{C}^B_{e,t})}{R_t}$. Similar to Lemma 3, we can derive that the computation delay $\hat{D}_{c,t}$ is a decreasing and convex function of the cloud computing resources $C_{c,t}$. Since $\hat{g}(C_{c,t})$ is an increasing and convex function of $C_{c,t}$, Problem **P2_1** is a convex optimization problem with respect to the cloud computing resources $C_{c,t}$. Thus, when $C_{c,t} < \bar{C}_{c,t}$, we can search for the optimal wholesale price $a_{2,t}$ and the optimal cloud computing resources $C_{c,t}$ by the dichotomy method comparing with the costs $p_{e,t}(a_{2,t})$ and $p_{c,t}(C_{c,t})$, sketched as steps 6–18 in Algorithm 2.

However, when the cloud computing resources $C_{c,t}$ reaches its upper bound $\bar{C}_{c,t}$, the cloud should select an optimal wholesale price $a_{2,t}$ to make a trade-off between the operation cost and the QoS penalty. Thus, the optimal wholesale price $a_{2,t}$ can be calculated by solving the following problem:

$$\begin{aligned} \textbf{P2\_2:} \min_{a_{2,t}} \quad & \sum_e (U^S_{e,t} - U^B_{e,t}) + U(\hat{D}_{c,t}) \\ s.t. \quad & a_{2,t} \geq 0, \quad \forall t, \\ & \hat{D}_{c,t} \leq \bar{D}_{c,t}, \quad \forall t. \end{aligned}$$

The objective function is to minimize the operation cost for buying computing resources from the MEC servers and the QoS penalty at the cloud while the constraints defining the ranges for the wholesale price $a_{2,t}$ and the computation delay $\hat{D}_{c,t}$.

According to Theorem 1, the available computing resources $\sum_e (C^C_{e,t} - \bar{C}^B_{e,t})$ is a non-decreasing and concave function of the wholesale price $a_{2,t}$. Since the computation delay $\hat{D}_{c,t}$ is a decreasing and convex function of total available computing resources $\hat{\mathbf{C}}_k$, $U(\hat{D}_{c,t})$ is a non-increasing and convex function of $a_{2,t}$. Thus, we proposed a heuristic algorithm based on the dichotomy method to find the optimal wholesale price $a_{2,t}$, sketched as steps 19–35 in Algorithm 2.

Algorithm 2 Optimal Pricing and Cloud Computing Resource Management

1 **Input:** $(\lambda_{e,t}, R_t, \bar{D}_{e,t})$ for computing tasks, $(C_{e,t}, K, T)$ for each MEC server, and $(\bar{C}_{c,t}, \bar{D}_{c,t})$ for the cloud;
2 **Output:** $\{a^*_{2,t}, C^*_{c,t}, \forall t\}$;
3 **for** *each time slot* t **do**
4 1) Set $a_{2,t} = p_{c,t}(\bar{C}_{c,t})$;
5 2) Calculate optimal $C^*_{c,t}$ by solving problem **P2_1**;
6 **if** $C^*_{c,t} < \bar{C}_{c,t}$, *Set* $\underline{a}_{2,t} = 0$ *and* $\overline{a}_{2,t} = a_{2,t}$, **then**
7 1) Set $a_{2,t} = \frac{\underline{a}_{2,t} + \overline{a}_{2,t}}{2}$;
8 2) Update optimal $C^*_{c,t}$ by solving problem **P2_1**;
9 **if** $p_{e,t}(a_{2,t}) < p_{c,t}(C^*_{c,t})$ **then**
10 Set $\underline{a}_{2,t} = a_{2,t}$ and go step 7;
11 **end**
12 **if** $p_{e,t}(a_{2,t}) > p_{c,t}(C^*_{c,t})$ **then**
13 Set $\overline{a}_{2,t} = a_{2,t}$ and go step 7;
14 **end**
15 **if** $p_{e,t}(a_{2,t}) = p_{c,t}(C^*_{c,t})$ **then**
16 $a^*_{2,t} = a_{2,t}$, Break;
17 **end**
18 **end**
19 **if** $C^*_{c,t} = \bar{C}_{c,t}$, *Set* $\underline{a}_{2,t} = a_{2,t}$, **then**
20 1) Set $a_{2,t} = (1+\epsilon)\underline{a}_{2,t}$ and update the cost $\mathbf{U}_{c,t}$;
21 **while** $\boldsymbol{U}_{c,t}|_{a_{2,t}} < \boldsymbol{U}_{c,t}|_{\underline{a}_{2,t}}$ **do**
22 Set $\underline{a}_{2,t} = a_{2,t}$ and update $\mathbf{U}_{c,t}|_{a_{2,t}}$ by step 20;
23 **end**
24 2) Set $\bar{a}_{2,t} = a_{2,t}$ when $\mathbf{U}_{c,t}|_{a_{2,t}} \geq \mathbf{U}_{c,t}|_{\underline{a}_{2,t}}$;
25 3) Set $a_{2,t} = \frac{\underline{a}_{2,t} + \overline{a}_{2,t}}{2}$ and update the cost $\mathbf{U}_{c,t}$;
26 **if** $\boldsymbol{U}_{c,t}|_{a_{2,t}} > \boldsymbol{U}_{c,t}|_{\underline{a}_{2,t}}$ **then**
27 Set $\overline{a}_{2,t} = a_{2,t}$ and go step 27;
28 **end**
29 **if** $\boldsymbol{U}_{c,t}|_{a_{2,t}} < \boldsymbol{U}_{c,t}|_{\underline{a}_{2,t}}$ **then**
30 Set $\underline{a}_{2,t} = a_{2,t}$ and go step 27;
31 **end**
32 **if** $\boldsymbol{U}_{c,t}|_{a_{2,t}} = \boldsymbol{U}_{c,t}|_{\underline{a}_{2,t}}$ **then**
33 $a^*_{2,t} = a_{2,t}$, Break;
34 **end**
35 **end**
36 Calculate the optimal $C^{C*}_{e,t}$ and $\hat{C}^{B*}_{e,k}$ by EWBS in [28].
37 **end**
38 where ϵ is a small-size step.

Convergence Analysis

When $C^*_{c,t} < \bar{C}_{c,t}$, based on the dichotomy method, the available range for the wholesale price $a_{2,t}$ will be narrowed by steps 7–14 until $p_{e,t}(a_{2,t}) = p_{c,t}(C^*_{c,t})$, which means that the optimal wholesale price $a^*_{2,t}$ has been obtained. Since the total operation cost will be decreased by each step, steps

TABLE I
THE NUMBER OF MEC SERVERS WITH DIFFERENT ARRIVAL RATES

$\lambda_{e,t}$	6	7	8	9	10	11	12
Low	20	10	5	2	1	1	1
Middle	4	5	6	8	6	5	6
High	1	1	2	2	4	10	20

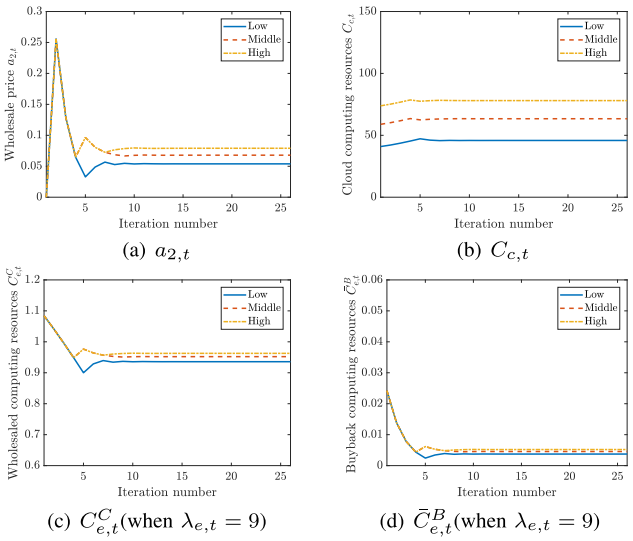
6–18 will converge to a unified solution. When $C^*_{c,t} = \bar{C}_{c,t}$, it means that the total operation cost maybe decreased by increasing the wholesale price $a_{2,t}$. Thus, steps 20–24 are used to find the range of the optimal wholesale price $a^*_{2,t}$. With the increase of the wholesale price $a_{2,t}$, the total operation cost will go to infinity. Thus, there exists a unified upper bound $\bar{a}_{2,t}$ to maximize the total profit of the cloud. Similarly, steps 25–34 will narrow the range of the optimal wholesale price $a^*_{2,t}$. Since the total operation cost will be decreased at each step, steps 19–34 will converge to a unified solution. Thus, the proposed algorithm 2 will converge to a unified solution.

VI. NUMERICAL EVALUATIONS

We evaluate the proposed social welfare maximization and the optimal pricing and cloud computing resource management (named optimal pricing scheme in the simulation section) in mobile edge-cloud computing networks and show numerical results and analysis in this section. The evaluation setting is given as follows: There are 40 MEC servers, each of which has $C_{e,t} = 3.2$GHz computing resources, and 10 cloud servers at the cloud with an upper bound on computing resources $\bar{C}_{c,t} = 200$GHz. The costs for cloud computing resources are $c_3 = \$0.02487/(\text{GHz*hour})$ and $c_4 = \$0.002487/(\text{GHz*hour})^2$. We assume that the arrival of computing tasks at MEC servers are located in $[6, 12]$ per second and that at the cloud is $\lambda_{c,t} = 500$ per second. To characterize the time-varying computing workloads at MEC servers, the specific distribution of $\lambda_{e,t}$ in low, middle and high statuses can be found in Table I. The expected computing workload of each computing task is 100Kb and each bit needs about 2000 cycles computing resources to be processed [42]. We assume that the service fees at the MEC servers and the cloud are the same, e.g., $a_{1,t} = \$0.2764/Gb$. The parameters for buying computing resources back are $c_1 = \$0.2736/(\text{GHz*hour})$ and $c_2 = \$0.8208/(\text{GHz*hour})^2$. The deadline for the computing tasks at MEC servers are $\bar{D}_{e,t} = 2s$ and that at the cloud is $\bar{D}_{c,t} = 4s$. The QoS penalty is $U(\hat{D}_{c,t}) = \$10^4\hat{D}_{c,t}$. To evaluate the performance of the proposed algorithm, we compare the simulation results with the following two schemes: 1) the MEC servers cannot share their computing resources to the cloud and may buy computing resources from the cloud to guarantee their QoS, named "no sharing"; 2) the wholesale price is a constant, i.e., $a_2 = \$0.2487/(\text{GHz*hour})$, named "constant price". Note that, for the social welfare maximization, we assume that the income for the MEC and the cloud is the service fee for them providing computing services and no profit of the cloud will be transferred to the MEC.

A. Simulation Results and Performance Analysis

The trajectories of the variables in the proposed algorithm are shown in Fig. 5. It can be found that the proposed algorithm will find the solution within 15 iterations. Furthermore,



(a) $a_{2,t}$ (b) $C_{c,t}$

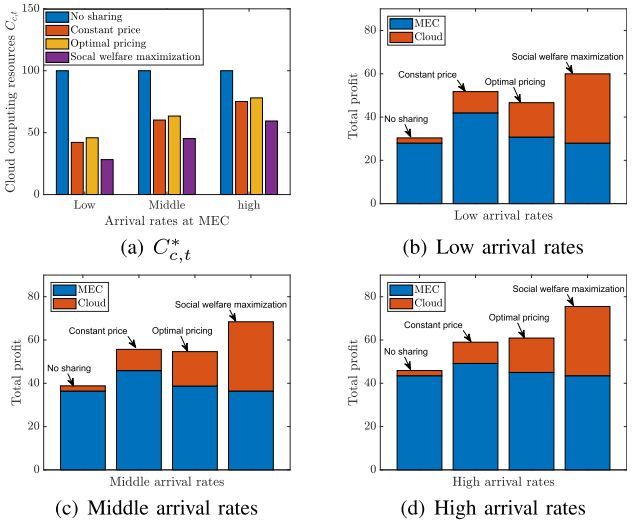
(c) $C^C_{e,t}$(when $\lambda_{e,t} = 9$) (d) $\bar{C}^B_{e,t}$(when $\lambda_{e,t} = 9$)

Fig. 5. The trajectories of the variables: (a) the wholesale price $a_{2,t}$, (b) the wholesaled computing resources $C_{c,t}$, and (c) the wholesaled and (d) the buyback computing resources $C^C_{e,t}$ and $\bar{C}^B_{e,t}$ for the MEC server with $\lambda_{e,t} = 9$.

with the increase of the computing tasks at the MEC servers, all of the parameters, including the wholesale price $a_{2,t}$, the cloud computing resource $C_{c,t}$, as well as the wholesaled computing resource $C^C_{e,t}$ and the expected buyback computing resources $\bar{C}^B_{e,t}$ (with given arrival rate of computing tasks), will increase. The reason can be summarized as follows: with the increase of the computing tasks at the MEC, 1) the MEC servers need to reserve more computing resources to satisfy their computing requirements and reduce the buyback cost; 2) for the cloud, since the wholesaled computing resources from the MEC servers decrease, it needs to increase the wholesale price and/or the cloud computing resources to make a good trade-off between the operation cost and the QoS penalty; 3) for the MEC server with $\lambda_{e,t} = 9$, since the wholesale price increases, it will wholesale more computing resources to the cloud to generate more profit and the buyback computing resources will increase to guarantee the QoS.

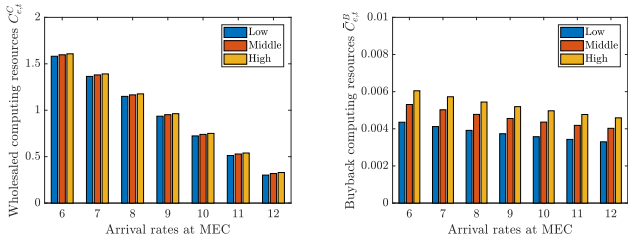
The optimal cloud computing resources $C^*_{c,t}$ under different schemes and the corresponding profits of the MEC and the cloud are shown in Fig. 6. From Fig. 6(a), it can be found that "no sharing" obtains the highest cloud computing resources since all the computing tasks at the cloud will be processed by the cloud computing resources while the social welfare maximization has the lowest cloud computing resources since all the available computing resources at the MEC servers will be utilized to reduce the local operation cost of the cloud. Furthermore, "constant price" has a lower computing resources than the optimal pricing scheme. This is because the wholesale price under "constant price" is $a_{2,t} = \$0.02487/(\text{GHz*hour})$, which is much higher than optimal wholesale prices $a^*_{2,t} = \{\$0.05388, \$0.06790, \$0.07908\}/(\text{GHz*hour})$, and thus "constant price" will obtain more computing resources from the MEC with a higher operation cost.

From Figs. 6(b)-6(d), it can be found that, with the increase of the arrival rates at the MEC servers, the total profits for all the schemes will increase. The social welfare maximization obtains the highest total profit and highest profit of the cloud since most of the computing resources are from the MEC without any profit transfer. Thus, the social welfare



(a) $C^*_{c,t}$ (b) Low arrival rates (c) Middle arrival rates (d) High arrival rates

Fig. 6. The cloud computing resources $C^*_{c,t}$ and the corresponding profit.



(a) $C^C_{e,t}$ (b) $\bar{C}^B_{e,t}$

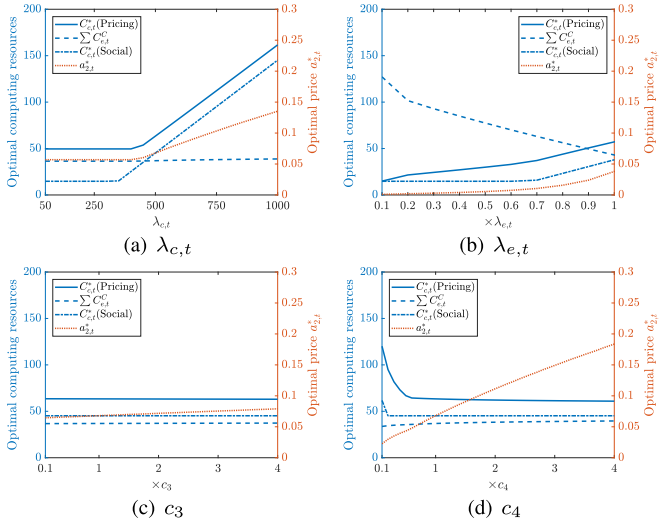
Fig. 7. The wholesaled computing resources $C^C_{e,t}$ and the buyback computing resources $\bar{C}^B_{e,t}$ for the MEC servers with different arrival rates.

maximization obtains the lowest profit of the MEC servers. Obviously, the "no sharing" obtain the lowest total profit and the lowest profit of the cloud since the cloud has a high operation cost for the cloud computing resources. "constant price" has a higher total profit than the optimal pricing scheme. However, "constant price" has a lower profit of the cloud than the optimal pricing scheme. Thus, the proposed optimal pricing scheme can improve the profitability of cloud, which is much preferred by the cloud. Furthermore, comparing with "no sharing" and the social welfare maximization, the optimal pricing scheme can improve the profitability of the MEC.

The wholesale and the buyback computing resources at the MEC servers are shown in Fig. 7. It can be found that, with the increase of total computing workload, the MEC server with a given arrival rate will wholesale more computing resources to the cloud to obtain more profit. To satisfy the computing requirements, the MEC server with a higher arrival rate will wholesale fewer computing resources to the cloud. Also, with the increase of total computing workload, the MEC server with a lower arrival rate will buy a little more computing resources back from the cloud since they can generate more profit by wholesaling more computing resources to the cloud. Note that, the buyback computing resource is very small compared with the wholesaled computing resources since the buyback cost is very large compared with the wholesale price.

B. Performance Analysis of System Parameters

Generally, the optimal decision, e.g., the optimal wholesale price $a^*_{2,t}$, the cloud computing resource $C_{c,t}$, and the



(a) $\lambda_{c,t}$ (b) $\lambda_{e,t}$ (c) c_3 (d) c_4

Fig. 8. The effects of system parameters on the optimal decision.

wholesaled and buyback computing resources $C^C_{e,t}$ and $\bar{C}^B_{e,k}$, is affected by several system parameters, e.g., the arrival of computing tasks at the cloud $\lambda_{c,t}$, the arrival of computing tasks at the MEC servers $\lambda_{e,t}$, the cost parameters for generating computing resources at the cloud, etc. In this paper, we analyze the effects of several system parameters on the optimal decision and show the results in Fig. 8. From these figures, it can be seen that the cloud computing resources $C^*_{c,t}$(Social) under the social welfare maximization is always smaller than $C^*_{c,t}$(pricing) under the optimal pricing scheme since more computing resources from the MEC can be utilized by the cloud under the social welfare maximization.

Effects of Computing Workload at the Cloud $\lambda_{c,t}$

As shown in Fig. 8(a), with the increase of computing workload $\lambda_{c,t}$ at the cloud, all of the variables, e.g., $C^*_{c,t}$ under the social welfare maximization and $C^*_{c,t}$, $a^*_{2,t}$ and $\sum C^C_{e,t}$ under the optimal pricing scheme, are non-decreasing. This is because the increasing computing workload $\lambda_{c,t}$ at the cloud may increase the demand of computing resources to achieve a good trade-off between the operation cost and the QoS penalty. It can be found that there exists a threshold for $\lambda_{c,t}$. When the computing workload $\lambda_{c,t}$ is below the threshold, $C^*_{c,t}$ under both the social welfare maximization and the optimal pricing scheme and the wholesale price $a^*_{2,t}$ will keep at their lowest levels due to the low QoS penalty. When the computing workload $\lambda_{c,t}$ exceeds the threshold, $C^*_{c,t}$ under the social welfare maximization and the optimal pricing scheme and the wholesale price $a^*_{2,t}$ under the optimal pricing scheme increase nearly linearly. Note that, the growth of the cloud computing resources $C^*_{c,t}$ is much larger than that of the wholesaling price $a^*_{2,t}$ and the wholesaled computing resources $\sum C^C_{e,t}$ due to the high buyback cost and QoS guarantee at the MEC.

Effects of Computing Workload at the MEC $\lambda_{e,t}$

As shown in Fig. 8(a), with the increase of computing workload $\lambda_{e,t}$ at MEC servers, the wholesaled computing resources $\sum C^C_{e,t}$ will decrease while the cloud computing resources $C^*_{c,t}$ and the wholesale price $a_{2,t}$ increase. This is because the MEC servers need to reserve more computing resources to guarantee

the QoS when the computing requirements are higher. Due to the decrease of the wholesaled computing resources $\sum C^C_{e,t}$, the cloud has to increase the cloud computing resource $C^*_{c,t}$ and the wholesale price $a^*_{2,t}$ to make a good trade-off between the operation cost and the QoS penalty. Thus, with the increase of $\lambda_{e,t}$, the operation cost at the cloud will be increased while the profit of the MEC will be increased, which have been shown in Fig. 6.

Effects of Cost Parameters c_3 and c_4

As shown in Figs. 8(c)-8(d), with the increase of cost parameters c_3 and c_4, the wholesaling price $a^*_{2,t}$ will increase and the effect of c_4 is much higher than that of c_3 due to the big value of $(C_{c,t})^2$. With the increase of c_3, the computing resources $C^*_{c,t}$ under both social welfare maximization and optimal pricing scheme and the wholesaled computing resource $\sum C^C_{e,t}$ remains basically the same since the cloud increases the wholesale price $a^*_{2,t}$ a little bit to make a good trade-off between the operation cost and the QoS penalty. With the increase of c_4, the wholesale price $a^*_{2,t}$ will increase quickly due to the high operation cost when the cloud computing resources $C^*_{c,t}$ is large. Note that, when the value of c_4 is very small, the cloud computing resources $C^*_{c,t}$ with optimal pricing scheme will be much higher due to the low cost parameters. Thus, building a cloud with a low c_4 can generate profit much easier.

VII. Conclusions

In this paper, we proposed an efficient framework for the MEC and the cloud to share their computing resources with each other to improve their profitabilities. We formulated the computing resource management for the MEC and the cloud as their respective profit maximization problems. Then, we solve these problems considering two cases: i) social welfare maximization by assuming that both the MEC and the cloud belong to the same entity and the computing resource sharing happens without profit transfers; ii) the respective profit maximization by assuming that the MEC and the cloud belong to different entities and the computing resource sharing only happens with profit transfers. For the first case, we proved that the social welfare only depends on the cloud computing resources and the concavity of the social welfare maximization problem. For the second case, we analyze the relationship between the wholesale price and the available computing resources from the MEC and then designed an optimal pricing and cloud computing resource management to maximize the total profit. Numerical evaluation shows that the proposed algorithms can maximize the social welfare and the respective profits of the MEC and the cloud separately. Furthermore, we analyze the effects of system parameters on the system performance and show that the cost parameter c_4 affects the wholesale price more substantially.

In this paper, we assume that all the MEC servers have the same computing resources and all the computing tasks with similar QoS requirements. In our future work, we will consider the computing resources sharing in large-scale mobile edge-cloud computing networks, in which there are several MEC servers belong to different entities with various computing resources and QoS requirements and several cloud networks to compete for the wholesale computing resources from the MEC servers simultaneously. The cloud can issue different prices to different entities to improve its benefit. Furthermore, we will take the communication delay between the MEC server and the cloud/other MEC servers, as well as the effects of energy consumption, into consideration to manage the computing resource sharing in mobile edge-cloud computing networks. Also, we will further discuss the design of admission control when the arrival of computing tasks exceeds the full capacity of the MEC servers and the cloud.

References

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[2] L. Cai, J. Pan, L. Zhao, and X. Shen, "Networked electric vehicles for green intelligent transportation," *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 77–83, Jun. 2017.

[3] Y. Li, K. Sun, and L. Cai, "Cooperative device-to-device communication with network coding for machine type communication devices," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 296–309, Jan. 2018.

[4] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, "Narrowband Internet of Things: Implementations and applications," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2309–2314, Dec. 2017.

[5] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–36, 2019.

[6] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2495–2508, Sep. 2018.

[7] N. Cheng *et al.*, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[8] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.

[9] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Sep./Oct. 2017.

[10] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[11] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[12] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[13] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile-edge computation offloading: Energy-efficient resource management," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7590–7605, Nov. 2018.

[14] 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.

[15] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

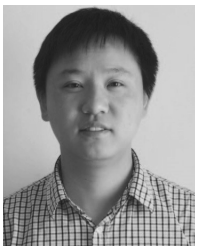
[16] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[17] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.

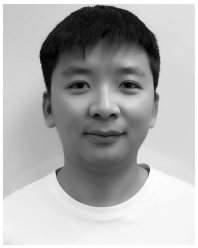
[18] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[19] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[20] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.
[21] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Online auctions in IaaS clouds: Welfare and profit maximization with server costs," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1034–1047, Apr. 2017.
[22] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.
[23] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[24] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov. 2018.
[25] G. Liu and H. Shen, "Minimum-cost cloud storage service across multiple cloud providers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2498–2513, Aug. 2017.
[26] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2118–2131, Oct. 2018.
[27] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
[28] Y. Zhang, X. Lan, Y. Li, L. Cai, and J. Pan, "Efficient computation resource management in mobile edge-cloud computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3455–3466, Apr. 2019.
[29] H. Mendelson and S. Whang, "Optimal incentive-compatible priority pricing for the M/M/1 queue," *Oper. Res.*, vol. 38, no. 5, pp. 870–883, Oct. 1990.
[30] WikiHow Staff. (Mar. 29, 2019). *How to Test Network and Internet Latency (Lag) in Microsoft Windows*. [Online]. Available: https://www.wikihow.com/Test-Network-and-Internet-Latency-(Lag)-in-Microsoft-Windows
[31] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin, "Smart-Edge-CoCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT," *IEEE Netw.*, vol. 33, no. 2, pp. 58–64, Mar. 2019.
[32] S. Sundar and B. Liang, "Offloading dependent tasks with communication delay and deadline constraint," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 37–45.
[33] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar. 2017.
[34] N. Jain and J. Lakshmi, "PriDyn: Enabling differentiated I/O services in cloud using dynamic priorities," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 212–224, Mar. 2015.
[35] I. Menache, A. Ozdaglar, and N. Shimkin, "Socially optimal pricing of cloud computing resources," in *Proc. 5th Int. ICST Conf. Perform. Eval. Methodologies Tools*, 2011, pp. 322–331.
[36] V. S. Prasad, B. Vh, and T. A. Koka, "Mathematical analysis of single queue multi server and multi queue multi server queuing models: Comparison study," *Global J. Math. Anal.*, vol. 3, no. 3, pp. 97–104, 2015.
[37] J. D. C. Little and S. C. Graves, "Little's law," in *Building Intuition*. Boston, MA, USA: Springer, 2008, pp. 81–100.
[38] W. Grassmann, "The convexity of the mean queue size of the M/M/c queue with respect to the traffic intensity," *J. Appl. Probab.*, vol. 20, no. 4, pp. 916–919, Dec. 1983.
[39] H. L. Lee and M. A. Cohen, "A note on the convexity of performance measures of M/M/c queueing systems," *J. Appl. Probab.*, vol. 20, no. 4, pp. 920–923, Dec. 1983.
[40] Y. Zhang, P. You, and L. Cai, "Optimal charging scheduling by pricing for EV charging station with dual charging modes," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3386–3396, Sep. 2019.
[41] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1632–1646, Jun. 2016.
[42] X. Hu, K.-K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.



Yongmin Zhang (Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2015. From 2015 to 2019, he was a Post-Doctoral Research Fellow at the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada. He is currently a Professor with the School of Computer Science and Engineering, Central South University, China. His research interests include resource management and optimization in wireless networks, smart grid, and mobile computing. He won the Best Paper Award of the IEEE PIMRC'12 and the IEEE Asia-Pacific Outstanding Paper Award 2018.



Xiaolong Lan received the B.S. degree in mathematics and applied mathematics from the Chengdu University of Technology and the Ph.D. degree in information and communication engineering from Southwest Jiaotong University, China, in 2012 and 2019, respectively. From 2017 to 2019, he was a visiting Ph.D. student at the University of Victoria, Victoria, BC, Canada. He is currently an Associate Researcher with the College of Cybersecurity, Sichuan University, Chengdu, China. His current research interests include physical layer security, buffer-aided communication, energy-harvesting wireless communication, and mobile edge computing.

Ju Ren (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Central South University, China, in 2009, 2012, and 2016, respectively, all in computer science. From 2013 to 2015, he was a visiting Ph.D. student at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University, China, and also an Adjunct Professor with the School of Computer Science and Engineering, Central South University, China. His research interests include the Internet-of-Things, wireless communications, network computing, and cloud computing. He won many best paper awards from the IEEE flagship conferences, including the IEEE ICC'19 and the IEEE HPCC'19, and so on, and the IEEE TCSC Early Career Researcher Award (2019).

Lin Cai (Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees (awarded Outstanding Achievement in graduate studies) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things. She is an NSERC E.W.R. Steacie Memorial Fellow. She was a recipient of the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively, and the Best Paper Awards of the IEEE ICC 2008 and the IEEE WCNC 2011. She has co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society Board of Governors (2019–2021).