HetMEC: Latency-Optimal Task Assignment and Resource Allocation for Heterogeneous Multi-Layer Mobile Edge Computing

Pengfei Wang, *Student Member, IEEE*, Zijie Zheng, *Student Member, IEEE*, Boya Di, *Member, IEEE*, and Lingyang Song, *Fellow, IEEE*

***Abstract*— Driven by great demands on low-latency services of the edge devices (EDs), mobile edge computing (MEC) has been proposed to enable the computing capacities at the edge of the radio access network. However, conventional MEC servers suffer some disadvantages such as limited computing capacity, preventing and computation-intensive tasks to be processed on time. To relief this issue, we propose the heterogeneous multi-layer MEC (HetMEC) where data that cannot be timely processed at the edge are allowed to be offloaded to the upper-layer MEC servers, and finally to the cloud center (CC) with more powerful computing capacity. We aim to minimize the system latency, i.e., the total computing and transmission time on all layers for the data generated by the EDs. We design the latency minimization algorithm by jointly coordinating the task assignment, computing, and transmission resources among the EDs, multi-layer MEC servers, and the CC. The simulation results indicate that our proposed algorithm can achieve a lower latency and higher processing rate than the conventional MEC scheme.**

***Index Terms*— Heterogeneous multi-layer mobile edge computing (MEC), task assignment, resource allocation.**

I. Introduction

With the rise of the Internet of Things (IoT), namely a network including interconnected devices capable of exchanging information [1], [2], huge amount of data are generated and transmitted throughout the communication networks [3]. However, the computing capacities of the current communication networks are not sufficient to satisfy users' increasing demands on high data rates [4]. Traditionally, cloud computing has been proposed as an effective solution for such data explosion by making use of the strong computing capacity of the data center [5]. As a centralized paradigm, cloud computing can provide a wide range of services and massive computing resources supported by a large group of computers in the data center. However, the data transmission from the edge of the network to the remote cloud center usually induces high latency, which is unacceptable for the latency-sensitive applications [6], [7].

To deal with the dilemma of cloud computing, mobile edge computing (MEC) has been investigated, which enables the computation to be performed at the mobile devices and the access points (APs)[1] within the radio access networks [8], [9]. The MEC servers that possess the computing resources, e.g., the APs, offer rich services in close proximity to the end users, also known as the edge devices (EDs) [10]. When these EDs generate computation tasks at the edge of the communication networks, they can offload tasks to the MEC servers nearby rather than the remote cloud center [11]. Therefore, the MEC provides the low-latency and high-efficient data processing due to the proximity of the computing resources [12], [13]. However, most works only consider the MEC servers that directly communicate with the EDs via wireless links to offer the in-proximity services [14]–[16]. Due to the limited computing capacities of these MEC servers, it would be desirable that the data that cannot be processed at the MEC servers can be further offloaded to the upper-layer MEC servers, until to the cloud center (CC).

In this paper, we consider *heterogeneous multi-layer MEC* (HetMEC) for uplink communications to reduce the overall system latency, i.e., the total computing and transmission time of all layers of MEC servers and the CC. In the HetMEC network, EDs divide and offload the computational intensive tasks to multi-layer MEC servers and the CC for latency performance improvement [6]. Classified by the locations in the wired-wireless networks, various function nodes with certain computing capacities serve as the *MEC servers* on different layers, that is, the APs, switches, network gateways, and small data centers from the bottom up [17]. The data flow of each ED first transits in the radio access networks via a wireless ED-AP link. The received data of each AP are then partially processed and delivered to the wired core network, passing through the switches, network gateways, and the small data centers sequentially [18]. Locating at bottom-up layers, these MEC servers provide increasing computing capacity for

Manuscript received January 16, 2019; revised May 12, 2019 and July 16, 2019; accepted July 22, 2019. Date of publication August 6, 2019; date of current version October 9, 2019. This work was supported by the National Nature Science Foundation of China under Grant 61625101. The associate editor coordinating the review of this article and approving it for publication was Z. Dawy. *(Corresponding author: Lingyang Song.)*

The authors are with the Department of Electronics, Peking University, Beijing 100871, China (e-mail: wangpengfei13@pku.edu.cn; zijie.zheng@pku.edu.cn; diboya@pku.edu.cn; lingyang.song@pku.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2931315

[1] Base stations are the typical APs in the radio access networks.

data processing and finally send the data from the bottom layers to the remote CC.[2] Based on such a HetMEC structure, the computing resources of multi-layer MEC servers and the CC can be fully exploited to support computation-intensive and latency-sensitive tasks of the EDs with strong robustness.

A number of challenges induced by the heterogeneous nature of the multi-layer MEC networks still remain to be solved. *First*, since the data of each task can be divided and partially processed by multiple MEC servers on different layers, the task assignments among these servers are coupled with each other by the limited resources of their own. In other words, the amount of offloaded data in one MEC layer is correlated with that in all the other layers, which is different from that in the traditional MEC networks.[3] *Second*, the transmission resource allocation in both the wireless and wired network need to be considered, which are closely related with the task assignment among multiple layers of MEC servers. To be specific, the allocated wired transmission resources directly restrict the data transmission rate between adjacent layers of MEC servers. *Third*, due to the limited computing capacity of each MEC server, the robustness of the HetMEC network should be considered and evaluated in response to various data generation speed at the EDs for different applications.

In the literature, the above challenges induced by the HetMEC architecture have not been fully addressed [20]–[22]. Most existing works either do not consider the task assignment, computing and transmission resource allocation jointly [23]–[27], or fail to depict the relations between multiple layers in the HetMEC network [28], [29]. In [23], an efficient k-out-of-n task assignment scheme is proposed to minimize the execution time on multiple processor nodes and save energy consumption. In [24], the transmission resource allocation is studied for multi-user mobile edge computational offloading constrained by the computation latency. Authors in [26] analyze the transmission latency and computation latency separately, taking the task assignment and computing rate control into account. In [28], the traditional MEC networks are discussed, where the MEC computing resource allocation and uplink power allocation are studied along with the binary[4] task offloading. In [29], the computation offloading and interference management are performed in the wireless cellular networks with a single MEC server. Unfortunately, joint task assignment among multi-layer MEC servers in the HetMEC network has not been taken into account together with the computing and transmission resource allocation.

The main contributions of our paper are summarized as follows.

- We study the HetMEC network consisting of the EDs, multiple layer of MEC servers and the CC. The uplink transmission is considered where the data generated at the EDs are processed at multiple layers of MEC servers and finally aggregated at the CC.
- In order to improve the latency performance of the HetMEC network, we jointly consider the task assignment, computing and transmission resource allocation among multiple layers. To minimize the sum of the computing and transmission time on all layers, a latency minimization algorithm (LMA) is developed to achieve the global-optimal system latency.
- Simulations are performed in the considered HetMEC networks with different numbers of layers, and the results show that our algorithm LMA achieves a lower latency and higher processing rate than the previous schemes. The influence of the number of MEC layers on the robustness performance has been discussed.

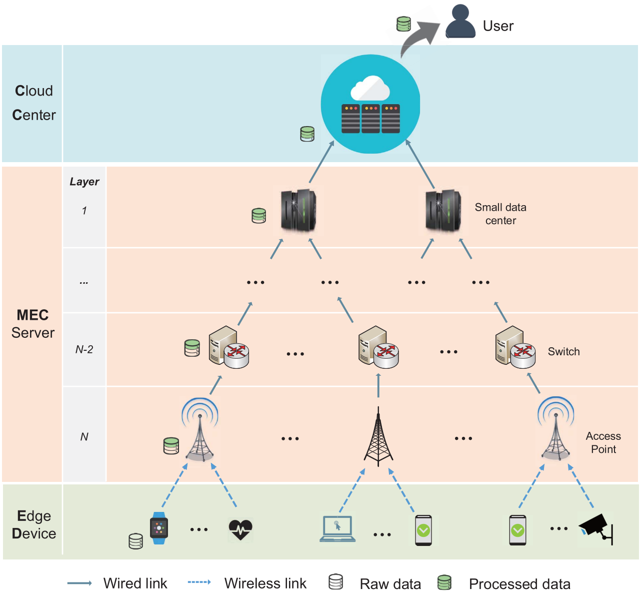


Fig. 1. The architecture of the HetMEC network.

The rest of the paper is organized as follows. In Section II we describe the system model of the HetMEC network. In Section III we discuss the system constraints of the HetMEC network and formulate the system latency minimization problem. To solve this problem, we design the algorithm LMA and analyze the influence of the number of MEC layers on the network robustness in Section IV. Simulation results are given in Section V. Finally the conclusions are drawn in Section VI.

II. System Model

As shown in Fig. 1, we consider a HetMEC network consisting of the EDs, N layers of MEC servers, and a CC. Each ED accesses the network by the AP through the wireless links between them. The uploaded data received by the EDs can then be forwarded to the MEC servers on the upper layers, connecting to the CC via wired links.[5] The number

[2] It is worth noting that the switches, network gateways or small data centers are not all necessary in the wired networks. The APs are possible to connect with the network gateways directly, and the network gateways may also connect with the remote cloud center.

[3] In the traditional MEC networks, EDs can offload data to only one layer of MEC servers, i.e., the APs.

[4] The binary task offloading means that the task is impartible and is processed at either the edge device or the MEC server.

[5] Though wireless technologies are studied to be potential for the backhual design [30], in our paper, we only consider the traditional backhuals where MEC servers and the CC are connected via wired links, i.e., optical fiber.

TABLE I
SUMMARY OF KEY NOTATION

Variable	Definition
N	Number of layers of the MEC servers
M_n	The number of devices on each layer n
Q_n^i	The number of child nodes connected with the parent node i on layer n
λ_{N+1}^i	Data generation speed of ED i
b_{N+1}^i	The required number of CPU cycles of ED i
$\lambda_n^{j,i}$	The raw data arrival speed at node j on layer n from its child node i
$b_n^{j,i}$	The required number of CPU cycles of node j on layer n for processing the data from its child node i
λ_n^j	The equivalent raw data arrival speed at node j on layer n
b_n^j	The required number of CPU cycles of node j on layer n
β_n^j	The total volume of the processed data received by node j on layer n
s_n^i	The (equivalent) task division percentage at node i on layer n
$s_n^{j,i}$	The task division percentage of node j on layer n for the data delivered from its child node i
θ_n^i	The computing capacity of node i on layer n
$\theta_n^{i,u}$	The maximum computing capacity of node i on layer n
$\phi_{n+1}^{j,i}$	The transmitting capacity of node i on layer $n+1$ to its parent node j
ϕ_n^j	The total transmission resources of node j on layer n

of devices on each layer n is denoted by M_n, $1 \leq n \leq N$. In such a tree structure, each node (ED or MEC server) connects with at most one parent node in the upper layer. The number of the child nodes[6] connected with the parent node i on layer n is denoted by Q_n^i, and the set of the child nodes is denoted by $\mathcal{Q}_n^i$. Each node has a different computing capacity.[7] To communicate with its child nodes, each MEC server and the CC possesses a certain amount of wireless or wired transmission resources.[8] We assume that all nodes access the wireless or wired channel via time division multiple access (TDMA) technology, implying that the frequency bands occupied by any two APs are orthogonal.

For a typical uplink MEC application, where the raw data are generated at the EDs and the results of the data processing need to be aggregated at the CC. The task generated at the ED is divided into multiple parts, and the ED, each MEC server on different layers or the CC only processes a part of it.[9] After processing its own part, the processing results and the remaining raw data are delivered to the upper layer. The percentage of the data to be processed at each node is adjustable. Moreover, once the data are processed at the edge of the network, i.e., at the ED or MEC servers, the output results, which are then forwarded to the CC, usually have a much smaller size than the raw data. The computing and transmitting at different devices are performed in parallel. The processing results and the raw data do not need to be transmitted simultaneously, i.e., the device can first transmit the raw data to the upper layers so that they can start their computation instead of waiting for the lower layer to finish processing its part.

We consider the data generated in one time period at the EDs. For computing, the computing capacity of a device is represented by the amount of data processed in one period. For transmitting, the transmission capacity of a device can be reflected by the total amount of the data that can be transmitted in that time period. When multiple devices transmit the data to the same MEC server or CC via the TDMA manner, the MEC server or CC will allocate the time resources to those devices according to the proportion of their transmitted data amount in the total received data. Therefore, we utilize the data transmission amount in each period to reflect the transmission resource.

The mathematical models of data processing and transmitting at the ED, MEC server and CC are listed below.

A. Edge Device

The EDs, including the cars, smartwatches, cameras, etc., are on the bottom of the HetMEC networks, and usually responsible for generating the raw data. For convenience, we refer to the EDs as layer $N+1$. Each ED processes part of the raw data, and delivers the results of the raw data together with the rest raw data to the node (AP) on the N-th layer via wireless link. Let s_{N+1}^i represent the task division percentage of ED i, which satisfies that

$$0 \leq s_{N+1}^i \leq 1. \tag{1}$$

Let λ_{N+1}^i denotes the data generation speed of ED i, and b_{N+1}^i denotes the required number of CPU cycles for processing these data in each time period. For the same kind of tasks, the data format and the processing procedure are same. Therefore, the required number of CPU cycles are proportional to the amount of raw data, and the proportion coefficient is denoted by μ, representing the average required number of CPU cycles for processing each bit of raw data. The compression ratio after the data processing is denoted by ρ. The computing capacity and the wireless transmitting capacity of ED i connected with AP j on the upper layer per period of time is denoted by θ_{N+1}^i and $\phi_{N+1}^{j,i}$, respectively. The computing data volume is

[6] That is to say, $\sum_{i=1}^{M_{n-1}} Q_i^{n-1} = M_n$.

[7] The computing capacity can be described by cycles per second.

[8] Among the MEC servers, the APs possess the wireless transmission resources, while the others possess the wired transmission resources.

[9] The EDs, MEC servers and the CC all can process the raw data.

limited by its computing capacity.

$$b^i_{N+1}s^i_{N+1} \leq \theta^i_{N+1}, \tag{2}$$

and the maximum computing capacity that ED i can offer is denoted by $\theta^{i,u}_{N+1}$, and thus,

$$\theta^i_{N+1} \leq \theta^{i,u}_{N+1}. \tag{3}$$

The transmitting data volume is restricted by the wireless transmitting capacity of ED i, which is closely related with the wireless transmission resources allocated by AP j.

$$\rho\lambda^i_{N+1}s^i_{N+1} + \lambda^i_{N+1}(1 - s^i_{N+1}) \leq \phi^{j,i}_{N+1}, \tag{4}$$

where $\rho\lambda^i_{N+1}s^i_{N+1}$ is the processing results, and $\lambda^i_{N+1}(1 - s^i_{N+1})$ represents the remaining raw data to transmit to AP j. The total transmitting data volume of all EDs connected with AP j is *linearly* constrained by the wireless transmission resources of node j on the N-th layer, denoted by ϕ^j_N, which can be expressed by

$$\sum_{i\in\mathcal{Q}^j_N} \phi^{j,i}_{N+1} \leq \phi^j_N. \tag{5}$$

Remark 1: The constraints in (5) can describe such wireless resources which influence the wireless data rate in a linear manner, e.g., the spectrum and time resources. The power and the antenna resources cannot be modeled in the similar way [31], which are left for the future works.

B. Mobile Edge Computing (MEC) Server

The MEC servers share the computing pressure of the EDs. Being the bottom layer of the MEC servers, the APs connect with EDs via wireless links and enable the EDs to access the wired networks. Other MEC servers, e.g., the switches and network gateways, receive the raw data and the processing results from APs. After processing part of the receiving raw data, the MEC server forwards the rest raw data together with the processing results to its parent node (upper-layer MEC server or CC).

When multiple MEC servers connect to the same upper-layer MEC server or belong to the same switch or bridge, the upper-layer MEC server can coordinate the connected MEC servers and allocate the transmission resources in a centralized way. Then transmission resources, e.g., the bandwidth or time, can be divided linearly to the multiple nodes [33].

As shown in Fig. 2, we consider the node j on the layer $n, 1 \leq n \leq N$, which is connected with the node k (MEC server or CC) on the layer $n-1$.

The raw data arrival speed from its child node i on the $(n+1)$-th layer can be expressed by

$$\lambda^{j,i}_n = \phi^{j,i}_{n+1} \cdot \frac{(1 - s^i_{n+1})\lambda^i_{n+1}}{(1 - s^i_{n+1} + \rho s^i_{n+1})\lambda^i_{n+1} + \beta^i_{n+1}}, \tag{6}$$

where $\phi^{j,i}_{n+1}$ is the total data arrival speed to node j from node i on the lower layer, and only part of it is the raw data arrival speed. The raw data volume transmitted to node j from its child node i is $\lambda^i_{n+1}(1 - s^i_{n+1})$, and the processed data volume is $\lambda^i_{n+1}\rho s^i_{n+1} + \beta^i_{n+1}$, where s^i_{n+1} is the equivalent task division percentage[10] of node i and β^i_{n+1} represents the volume of the processed data under the $(n+1)$-th layer. Since the required number of CPU cycles are proportional to the amount of raw data for the same kind of tasks, the required number of CPU cycles of node j on layer n for processing the data from its child node i is expressed by $b^{j,i}_n = \mu\lambda^{j,i}_n$. Hence, the equivalent raw data arrival speed at node j on the n-th layer is denoted by

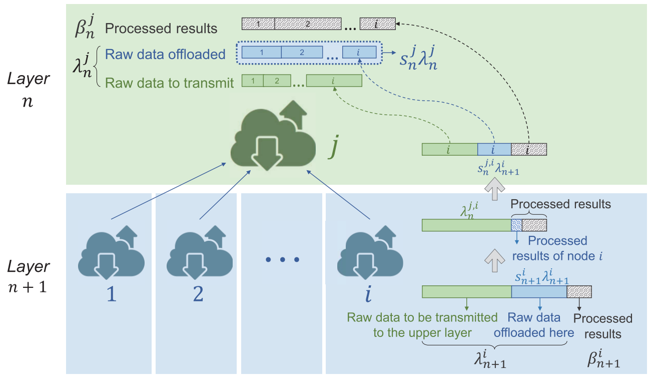


Fig. 2. The data processing and transmitting between two adjacent MEC layers of the HetMEC network.

$$\lambda^j_n = \sum_{i\in\mathcal{Q}^j_n} \lambda^{j,i}_n. \tag{7}$$

The required number of CPU cycles of node j on the n-th layer is denoted by $b^j_n = \mu\lambda^j_n$. Accordingly, the total volume of the processed data received by node j on the n-th layer can be expressed by

$$\beta^j_n = \sum_{i=1}^{N_j} \left(\beta^i_{n+1} + \rho \cdot s^i_{n+1}\lambda^i_{n+1}\right). \tag{8}$$

Let $s^{j,i}_n$ denotes the task division percentage of node j for the data delivered from node i on the $(n+1)$-th layer, which satisfies that

$$0 \leq s^{j,i}_n \leq 1, i \in \mathcal{Q}^j_n. \tag{9}$$

The computing capacity of node j is denoted by θ^j_n. The total CPU cycles required per second at node j, denoted by C^j_n, is limited by its computing capacity, which can be expressed by

$$C^j_n = \sum_{i\in\mathcal{Q}^j_n} s^{j,i}_n b^{j,i}_n \ \leq \ \theta^j_n. \tag{10}$$

Constrained by the limited computing resources, the maximum computing capacity that node j on the n layer can offer is denoted by $\theta^{j,u}_n$, and thus,

$$\theta^j_n \leq \theta^{j,u}_n. \tag{11}$$

Let $\phi^{k,j}_n$ denote the wired transmitting capacity of node j to node k. The transmitting data volume of node j is limited by

[10] When the node i is not at the bottom layer, i.e., it is not an ED, it may receives raw data from multiple links. s^i_{n+1} is the percentage of the total raw data volume to be processed at node i in its total received raw data volume.

its wired transmitting capacity, which is closely related with the wired transmission resources allocated by its parent node k on the upper layer.

$$\rho\lambda_n^j s_n^j + \lambda_n^j(1 - s_n^j) + \beta_n^j \le \phi_n^{k,j}. \qquad (12)$$

The data to be transmitted to node k on the upper layer includes three parts. $\rho\lambda_n^j s_n^j$ is the processed data of node j, and $\lambda_n^j(1 - s_n^j)$ is the remaining raw data to transmit, and β_n^j is the processed data delivered from the lower layer. All the three parts need to be transmitted to the upper layer, which is limited by the allocated wired transmitting capacity $\phi_n^{k,j}$ of node j. Moreover, the total transmitting data volume of all nodes on the n layer connected with the node k is limited by the wired transmission resources of the node k, denoted by ϕ_{n-1}^k, which can be expressed by

$$\sum_{j \in \mathcal{Q}_{n-1}^k} \phi_n^{k,j} \le \phi_{n-1}^k. \qquad (13)$$

C. Cloud Center

The CC collects the data from the MEC servers via wired links. All raw data delivered to the CC is processed and the whole results are forwarded to the user who generates the task. For convenience, we refer to the CC as layer 0.

The equivalent raw data arrival speed at the CC can be calculated by

$$\lambda_0^1 = \sum_{i=1}^{M_0^1} \left[\phi_1^{1,i} \cdot \frac{(1 - s_1^i)\lambda_1^i}{(1 - s_1^i + \rho s_1^i)\lambda_1^i + \beta_1^i} \right], \qquad (14)$$

and the required number of CPU cycles of the CC is represented by $b_0^1 = \mu\lambda_0^1$. The arriving data at the CC includes three part: the remaining raw data, the processing results of the MEC servers and the processing results of the EDs. The raw data arrival speed is proportional to the remaining raw data volume percentage in the arriving data. Moreover, the computing capacity of the CC is denoted by θ_0^1, and the maximum computing capacity the CC can offer is denoted by $\theta_0^{1,u}$.

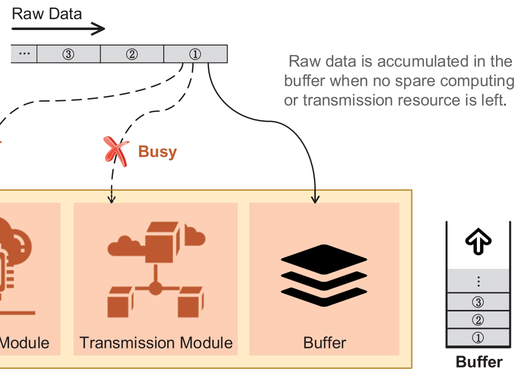
During the processing and transmitting from the EDs to the CC, the task assignment strategy $\boldsymbol{s}$, the computing capacity of each MEC server j on the n-th layer θ_n^j, the computing capacity of the CC θ_0^1, the wireless transmission resources allocation $\phi_{N+1}^{j,i}$ and the wired transmission resources allocation of the n-th layer, $\phi_n^{k,j}$, need to be optimized, which will be discussed in Section III.

III. Problem Formulation

In this section, we first analyze the system constraints of the considered HetMEC network, and then formulate the system latency minimization problem given these constraints.

A. System Constraints

We first describe when and why a HetMEC network can be out of function due to the traffic congestion. The total computing capacity of each node, the total wireless transmission resources of each AP, and the total wired transmission resources of each MEC server or the CC in our framework are finite, however, the data generation speed is fluctuant and time-varying. As shown in Fig. 3, when the data generation speed of the EDs exceeds a certain bound, the HetMEC network cannot follow up the data generation speed due to lack of available computing or transmission resources, and thus, the data will accumulate in the buffer.[11] As the raw data keep accumulating, the waiting time in the buffer increase, eventually leading to a congested network.



Raw data is accumulated in the buffer when no spare computing or transmission resource is left.

Fig. 3. An illustration of the congestion.

We then derive the system constraints of the HetMEC network under which the above congestion does not happen. Specifically, the data accumulation does not appear on each layer of the HetMEC network. In the N-layer HetMEC network, i.e., the HetMEC network with N layers of MEC servers, the execution of tasks is related to the computing of the EDs, N layers of MEC servers and the CC, as well as the transmitting of the EDs and N layers of MEC servers.

1) Constraints of the n-th Layer: We consider the n-th layer,[12] $1 \le n \le N+1$. After receiving the raw data and the results, the nodes on the n-th layer need to process a part of the raw data, and transmit the rest raw data together with the processing results to the upper layer. It is worth noting that the transmission resource allocated to each node on the n-th layer is determined by its parent node on the $(n-1)$-th layer. We consider the case that all nodes on the n-th layer fully use their computing capacity, which is the case when the equality holds in the following constraint.

$$b_n^i s_n^i \le \theta_n^i \le \theta_n^{i,u}, \quad \forall 1 \le i \le M_n, \ 1 \le n \le N+1, \qquad (15)$$

which implies that the transmission pressure of the n layer is minimum. Under the aforementioned circumstance, for each parent node j on the $(n-1)$-th layer, the total volume of the data to transmit from the n-th layer to the parent node j cannot surpasses the total transmission capacity of the parent node j. Hence, when $n = N+1$ (the EDs), the constraints for transmitting are described in (4) and (5). When $1 \le n \le N$ (the MEC servers), the constraints for transmitting are described in (12) and (13).

[11] The space of the buffer in each node is viewed as infinite, that is, the data only accumulates in the buffer and no data loss happens when facing the congestion.

[12] This layer may consist of the EDs or MEC servers. The constraints of them are similar.

2) Constraints of the CC: The CC needs to process all the remaining raw data delivered from the lower layer, and does not need to transmit. The volume of the arrived raw data at the CC per second should not surpass its computing capacity. Hence, the constraint of the CC is expressed by

$$b_0^1 \leq \theta_0^1. \tag{16}$$

Summarizing the constraints for all the layers of the HetMEC network, we have Proposition 1 to clarity the system constraints.

Proposition 1: The system constraints of the HetMEC network can be described by the constraints of each layer in the HetMEC network, i.e., the constraints that (1), (4), (5), (9), (12), (13), (15) and (16) are all satisfied.

B. Latency Minimization Problem Formulation

We aim to minimize the system latency of the HetMEC network, which is a general objective in the MEC networks [8], [15], [16], [26]. We first define the system latency in the HetMEC network and formulate the latency minimization problem in this subsection.

The latency of a task is defined as the total computing time and transmitting time from the ED to the CC. Consider the n-th layer where nodes receive the tasks from the lower layer. The nodes need to process the assigned raw data from the lower layer, and deliver the processing results as well as unprocessed data to the upper layer. Therefore, the total latency of all nodes on the n-th layer can be expressed by

$$L_n = \sum_{j=1}^{M_{n-1}} \sum_{i \in \mathcal{Q}_{n-1}^j} \left[\frac{s_n^i b_n^i}{\theta_n^i} + \frac{\rho s_n^i \lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i}{\phi_n^{j,i}} \right], \tag{17}$$

where $s_n^i b_n^i / \theta_n^i$ represents the processing time for the offloaded raw data, and $[\rho s_n^i \lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i]/\phi_n^{j,i}$ implies the transmitting time of the node i on the n-th layer.

We then define the system latency as below.

Definition 1: From the perspective of data processing, the ***system latency*** *is the total computing and transmission time of the EDs, all layers of MEC servers and the CC.*

Therefore, the system latency can be expressed as

$$L = \frac{b_0^1}{\theta_0^1} + \sum_{n=1}^{N+1} L_n, \tag{18}$$

where b_0^1/θ_0^1 represents the computing time of the CC.

Hence, the total latency minimization problem in the HetMEC network can be formulated as below.

$$\min_{\mathbf{s},\boldsymbol{\theta},\boldsymbol{\phi}} \quad L, \quad (1), (4), (5), (9), (12), (13), (15), (16). \tag{19}$$

IV. Latency Minimization Algorithm Design

In this section, we propose a latency minimization algorithm (LMA) to solve problem (19) via joint task assignment, computing and transmission resource allocation. We then analyze the influence of the number of MEC layers on the network robustness.

A. Algorithm Design

The system latency minimization problem described in (19) is nonconvex, in which the task assignment strategy $\mathbf{s}$, computing capacity allocation $\boldsymbol{\theta}$ and transmission resources allocation $\boldsymbol{\phi}$ are coupled. By utilizing the Cauchy-Schwarz inequality [34], we can obtain the following inequations.

$$\begin{aligned} & L(\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{\phi}) \\ & = \frac{b_0^1}{\theta_0^1} + \sum_{n=1}^{N+1} \sum_{j=1}^{M_{n-1}} \sum_{i \in \mathcal{Q}_{n-1}^j} \\ & \quad \times \left[\frac{s_n^i b_n^i}{\theta_n^i} + \frac{\rho s_n^i \lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i}{\phi_n^{j,i}} \right] \\ & \geq L_{min}(\boldsymbol{s}) = \left[\frac{b_0^1}{\theta_0^{1,u}} + \sum_{n=1}^{N+1} \sum_{j=1}^{M_{n-1}} \sum_{i \in \mathcal{Q}_{n-1}^j} \frac{s_n^i b_n^i}{\theta_n^{i,u}} \right] \\ & \quad + \sum_{n=1}^{N+1} \sum_{j=1}^{M_{n-1}} \frac{\left(\sum_{i \in \mathcal{Q}_{n-1}^j} \sqrt{\rho s_n^i \lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i} \right)^2}{\phi_{n-1}^j}, \end{aligned} \tag{20}$$

where $\theta_n^{i,u}$ and ϕ_{n-1}^j are the boundary of the computing and transmitting capacity.

Proposition 2: The task assignment strategy and resource allocation optimization can be separated in the proportional optimization problem (19) by utilizing the Cauchy-Schwarz inequality.

Proof: See Appendix A. ■

Proposition 3: The optimal computing capacity division and transmission resources allocation can be derived by the following relations.

$$\begin{aligned} & \frac{\phi_n^{j,i}}{\phi_n^{j,i'}} = \frac{\sqrt{\rho s_n^i \lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i}}{\sqrt{\rho s_n^{i'} \lambda_n^{i'} + (1-s_n^{i'})\lambda_n^{i'} + \beta_n^{i'}}}, \\ & \quad \theta_0^1 = \theta_0^{1,u}, \theta_n^i = \theta_n^{i,u}, \\ & \qquad \forall 1 \leq n \leq N+1, 1 \leq j \leq M_n, i, i' \in \mathcal{Q}_{n-1}^j. \end{aligned} \tag{21}$$

Proof: According to (20), the system latency $L(\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{\phi})$ achieves a minimum $L_{min}(\boldsymbol{s})$ when the equality holds. Based on the Cauchy-Schwarz inequality [34], the equality holds if and only if the equations in (21) are satisfied, implying that the computing capacity division $\boldsymbol{\theta}$ and transmission resources allocation $\boldsymbol{\phi}$ can be derived from the task assignment percentage $\boldsymbol{s}$. ■

Proposition 4: The objective function $L_{min}(\boldsymbol{s})$ in the task assignment problem (22) is concave, and the optimal results $\boldsymbol{s}^*$ are at the vertex of the feasible set bounded by the constraints.

$$\begin{aligned} & \min_{\boldsymbol{s}} \quad L_{min}(\boldsymbol{s}), \\ & s.t. \quad (1), (4), (5), (9), (12), (13), (15), (16), (21). \end{aligned} \tag{22}$$

Proof: See Appendix B ■

According to the aforementioned propositions, we can obtain the optimal results of the latency minimization problem by searching all the vertices of the feasible set bounded by the constraints. The latency minimization algorithm is summarized in **Algorithm 1**.

Algorithm 1 Latency Minimization Algorithm

```
Input: Computing capacity $\theta_n^i$, upper bound of the transmission resource of each node $\phi_n^{i,0}$, data generation speed $\boldsymbol{\lambda}$.
Output: Task assignment strategy $\boldsymbol{s}^*$, resources allocation scheme $\boldsymbol{\theta}^*, \boldsymbol{\phi}^*$.
for all Vertex of the feasible set do
    Obtain the corresponding task assignment strategy $\boldsymbol{s}$.
    Obtain the resource allocation scheme $\boldsymbol{\theta}, \boldsymbol{\phi}$ according to $\boldsymbol{s}$ and (21) in Proposition 3.
    if $L_{min}(\boldsymbol{s}) < L_{min}(\boldsymbol{s}^*)$ then
        $L_{min}(\boldsymbol{s}^*) = L_{min}(\boldsymbol{s})$.
        Update the optimal $\boldsymbol{s}^*, \boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$ with $\boldsymbol{s}^* = \boldsymbol{s}, \boldsymbol{\theta}^* = \boldsymbol{\theta}, \boldsymbol{\phi}^* = \boldsymbol{\phi}$.
```

We first convert the proportional optimization problem in (19) into the task assignment problem in (20) given **Proposition 3** by utilizing Cauchy-Schwarz inequality. Since the converted task assignment problem is proved concave according to **Proposition 4**, we calculate the system latency at each vertex of the feasible set bounded by the non-congested constraints, i.e., $L_{min}(\boldsymbol{s})$, where $\boldsymbol{s}$ is determined by the constraints associated with the discussing vertex. The computing and transmission resource allocation are also determined once the task assignment strategy $\boldsymbol{s}$ is fixed. After considering all the vertices, the minimum system latency together with the optimal task assignment strategy and resource allocation can be obtained.

Remark 2: Assuming the number of devices in the whole network is denoted by M, the complexity of the latency minimization algorithm is proportional to the number of the vertices of the feasible set. The maximum number of the vertices of the feasible set is in the square magnitude of the number of system constraints, i.e., $O(M^2)$.

Proof: See Appendix C. ■

We then elaborate on the implementation of our approaches in practice. To perform the LMA in the HetMEC network, the EDs and MEC servers need to register to the CC and upload the node information (i.e., the location, the amount of computing and transmission resource, the data generation speed). Based on the registration information, the CC then establish the logical graph of the HetMEC network. After the CC obtains the optimal task assignment and resource allocation scheme by performing our LMA algorithm, the CC needs to broadcast the scheme to all devices corresponding to the task. For the HetMEC network with more layers of MEC servers, the overhead of the registration and scheme broadcasting is larger, which is proportional to the number of devices. The network robustness is improved at the expense of the overhead for communication.

B. Network Robustness Analysis

In this subsection, we discuss the relation between the network robustness and the number of MEC layers of the HetMEC network, influenced by the amount of computing and transmission resources.

The network robustness is reflected by the network processing capacity, which is defined by the maximum data generation speed at the ED supported by the non-congested HetMEC networks. The ED can start processing the new arrival task once it finishes processing its own part of the previous task, without waiting it to be processed completely and transmitted to the CC. Intuitively, a network is more robust when more resources are available brought by a newly added layer. However, this may only be true when the added layer is selected properly, as will be analyzed in both computing and transmission resource shortage cases as below.

1) Computing Resource Shortage Case: As shown in Fig. 4, the bottleneck of the network processing capacity lies in the limited computing resources, which can be expressed by

$$b_n^i s_n^i = \theta_n^{i,u}, \quad \forall 1 \le n \le N, 1 \le i \le M_n, \tag{23}$$

$$\sum_{j \in \mathcal{Q}_{n-1}^k} \left(\rho \lambda_n^j s_n^j + \lambda_n^j (1 - s_n^j) + \beta_n^j\right) < \phi_{n-1}^k, \quad \forall 2 \le n \le N, 1 \le k \le M_{n-1}. \tag{24}$$

In this case, the computing resources of all layers are fully utilized, while there still remains the idle transmission resources in the HetMEC network. The network will be in congestion if the data generation speed λ continues to increase.

When adding a layer of MEC servers between the $(n_0 - 1)$-th and n_0-th layer of the initial network,[13] in order to increase the network robustness, the computing and transmission resources of the new added layer should satisfy that

$$\theta_{n_0}^{i,u} > 0, \quad \forall 1 \le i \le M_{n_0}, \tag{25}$$

$$\sum_{j \in \mathcal{Q}_{n_0}^i} \left(\rho \lambda_{n_0+1}^j s_{n_0+1}^j + \lambda_{n_0+1}^j (1 - s_{n_0+1}^j) + \beta_{n_0+1}^j\right) < \phi_{n_0}^i, \quad \forall 1 \le i \le M_{n_0}, \tag{26}$$

where each node on the added layer possesses the computing resources, and the transmission resources of each node on the added layer are sufficient enough to transmit all the data from the lower layer (i.e., $(n_0 + 1)$-th layer in the $(N + 1)$-layer network). Therefore, the added layer can relieve the processing pressure of the other layers. As the data generation speed λ continues to increase, the task division percentage s_n^i, $1 \le i \le M_n$, on any other layer, $n \ne n_0$, can be reduced, and the task division percentage $s_{n_0}^i, 1 \le i \le M_{n_0}$, increases until the following conditions are satisfied.

$$\frac{b_{n_0}^i s_{n_0}^i}{\theta_{n_0}^i} = \frac{b_n^j s_n^j}{\theta_n^j}, \quad \forall n \ne n_0, \; 1 \le i \le M_{n_0}, 1 \le j \le M_n, \tag{27}$$

$$\lambda_{n_0}^i s_{n_0}^i \le \theta_{n_0}^{i,u}, \forall 1 \le i \le M_{n_0}, \tag{28}$$

$$\sum_{j \in \mathcal{Q}_{n-1}^k} \left(\rho \lambda_n^j s_n^j + \lambda_n^j (1 - s_n^j) + \beta_n^j\right) < \phi_{n-1}^k, \quad \forall 2 \le n \le N + 1, 1 \le k \le M_{n-1}. \tag{29}$$

[13] The added layer becomes the n_0-th layer, and the initial n_0-th layer becomes the $(n_0 + 1)$-th layer.

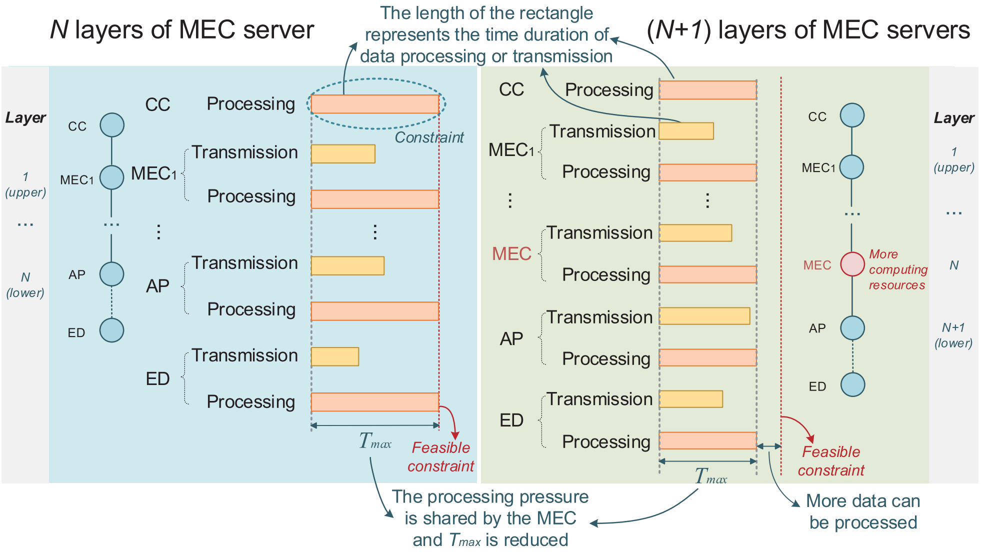

N layers of MEC server
The length of the rectangle represents the time duration of data processing or transmission
(N+1) layers of MEC servers
Layer
1 (upper)
N (lower)
CC
MEC1
AP
ED
Processing
Transmission
Constraint
Tmax
Feasible constraint
MEC
More computing resources
N+1 (lower)
The processing pressure is shared by the MEC and Tmax is reduced
More data can be processed

Fig. 4. The computing resource shortage case.

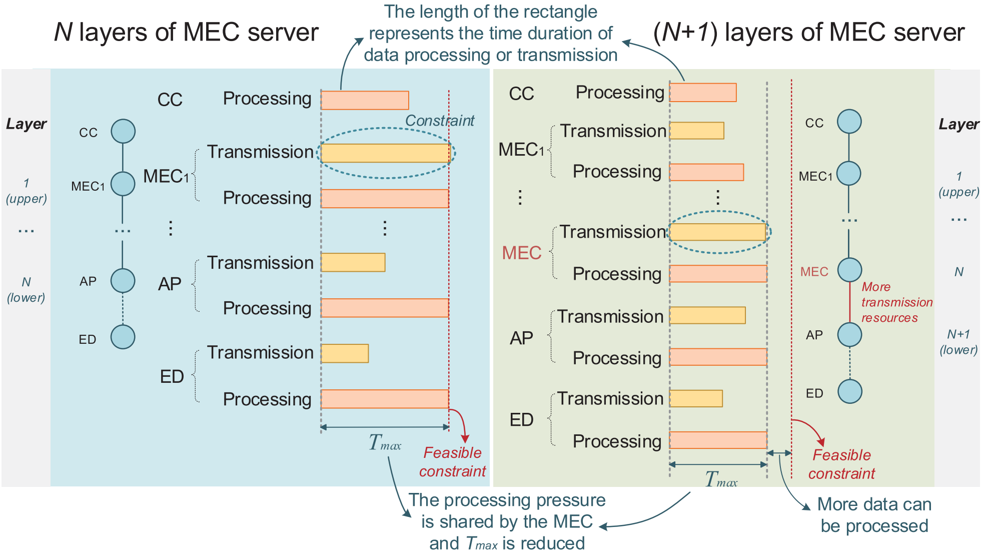

N layers of MEC server
The length of the rectangle represents the time duration of data processing or transmission
(N+1) layers of MEC server
Layer
1 (upper)
N (lower)
CC
MEC1
AP
ED
Processing
Transmission
Constraint
Tmax
Feasible constraint
MEC
More transmission resources
N+1 (lower)
The processing pressure is shared by the MEC and Tmax is reduced
More data can be processed

Fig. 5. The transmission resource shortage case.

The computing resources of the whole HetMEC network are then fully utilized where the processing time on all layers are equal and limited by the non-congested computing constraint.

2) Transmission Resource Shortage Case: As shown in Fig. 5, the bottleneck of the network processing capacity lies in the transmission resources of one layer n_0, and the layer is determined by the conditions described in (4), (5), (12) and (13). The transmission resources of the n_0-th layer have been fully utilized for the data transmission from the (n_0+1)-th layer to the n_0-th layer, which can be expressed by

$$b_n^i s_n^i < \theta_n^{i,u}, \quad \forall 0 \le n \le N+1, 1 \le i \le M_n, \tag{30}$$

$$\sum_{j\in\mathcal{Q}_{n_0}^k} \left(\rho\lambda_{n_0+1}^j s_{n_0+1}^j + \lambda_{n_0+1}^j (1-s_{n_0+1}^j) + \beta_{n_0+1}^j\right) = \phi_{n_0}^k, \quad \forall 1 \le k \le M_{n_0+1}. \tag{31}$$

The network will be congested between the (n_0+1)-th layer and the n_0-th layer if the data generation speed λ continues to increase.

The robustness can be enhanced only when adding a layer of MEC servers between the (n_0+1)-th layer and the n_0-th layer, or below the (n_0+1)-th layer. It does not make any contribution to the robustness enhancement to add a layer when the layer number is smaller than n_0 (i.e., the added layer is above the n_0-th layer). This is because the transmission resources are allocated by the parent node on the upper adjacent layer, and the operation of adding a layer of MEC servers above the n_0-th layer cannot increase the transmission resources of the n_0-th layer or reduce the amount of data transmitted from the (n_0+1)-th layer.

When adding a layer of MEC servers below the initial n_0-th layer, denoted by the n'-th layer,[14] in order to

[14] The added layer becomes the n'-th layer, and the initial n'-th layer becomes the $(n'+1)$-th layer, and $n' > n_0$.

increase the network robustness, the computing and transmission resources of the added layer should satisfy that

$$\theta_{n'}^{i,u} > 0, \quad \forall 1 \le i \le M_{n'}, \tag{32}$$

$$\sum_{j\in\mathcal{Q}_{n'}^{i}} \left(\rho\lambda_{n'+1}^{j} s_{n'+1}^{j} + \lambda_{n'+1}^{j}(1-s_{n'+1}^{j}) + \beta_{n'+1}^{j}\right) < \phi_{n'}^{i}, \quad \forall 1 \le i \le M_{n'}, \tag{33}$$

where each node on the added layer possesses the computing resources, and the transmission resources are sufficient enough to transmit all the data from the lower layer.

Tasks with a larger data generation speed can be processed since condition (31) has changed to the condition that the amount of the transmitted data from the (n_0+1)-th layer is strictly smaller than the transmission resources. Therefore, the network can remain non-congested when the data generation speed λ continues to increase. Condition (31) changes in two cases:

- If a layer is added between the (n_0+1)-th layer and the n_0-th layer, condition (31) changes because the transmission resources are more abundant.[15] Therefore, the amount of data to be transmitted from the (n_0+1)-th layer satisfies that

$$\sum_{j\in\mathcal{Q}_{n_0}^{k}} \left[\rho\lambda_{n_0+1}^{j} s_{n_0+1}^{j} + \lambda_{n_0+1}^{j}(1-s_{n_0+1}^{j}) + \beta_{n_0+1}^{j}\right] < \phi_{n_0+1}^{k}.$$

- If a layer is added below the (n_0+1)-th layer, condition (31) changes because the amount of data to transmit is reduced.[16] We assume that λ_0 volume of data that originally being processed on the (n_0+1)-th layer are offloaded to the added layer. The amount of data to be transmitted from the (n_0+1)-th layer is reduced and satisfies that

$$\sum_{j\in\mathcal{Q}_{n_0}^{k}} \left(\rho\lambda_{n_0+1}^{j} s_{n_0+1}^{j} + \lambda_{n_0+1}^{j}(1-s_{n_0+1}^{j}) + \beta_{n_0+1}^{j}\right) - \sum_{j\in\mathcal{Q}_{n_0}^{k}} \left((1-\rho)(1-s_{n_0+1}^{j})\lambda_0\right) < \phi_{n_0+1}^{k}.$$

We summarize the relation between the network robustness and the number of MEC layers influenced by the amount of computing and transmission resources. In the computing resource shortage case, the network robustness can be enhanced if the MEC servers which can provide the computing capacity on the new added layer satisfies (25) and (26). In the transmission resource shortage case, the network robustness can be enhanced only if a layer of MEC servers satisfying (32) and (33) are added *below* the initial transmission resource constrained MEC layer. When additional layers of MEC servers are added, to support a larger data generation speed at the EDs, the locations of these MEC servers need to be determined according to the current resource states of the network. Moreover, the trade-off between the latency and cost needs to be considered. Furthermore, due to the changed topology of considered devices, the task assignment, computing and transmission resource allocation among different layers need to be updated according to our proposed algorithm LMA.

TABLE II
HetMEC Network Parameters

Parameters	Value
The volume of the data file	60 Kbits
Compression ratio ρ	10%
The period of data generation	1s

We then evaluate the incurred overhead of considering more MEC layers. Assume that the amount of information that each device reports to the CC (e.g., the location, the amount of computing and transmission resources, the data generation speed) is a_1, and the amount of CC broadcast information (task assignment and resource allocation scheme) is a_2. Adding a layer of MEC servers implies that this MEC layer is considered in the data processing, and the physical connections of the CC-MEC network remains the same.

When M_n MEC servers are added on layer n, they need to register the node information to the CC through $(n-1)$ layers of MEC servers. Since the registration information are delivered to the CC layer by layer, the added overhead during the node registration is $O_1 = a_1 \cdot M_n \cdot n$. After performing the LMA at the CC, the CC need to broadcast the adjusted task assignment and resource allocation scheme to the other devices. The added overhead during the CC broadcasting is $O_1 = a_2 \cdot M_n$. Therefore, the incurred overhead of adding M_n MEC servers on layer n is expressed by

$$O(n, M_n) = O_1 + O_2 = (a_1 \cdot n + a_2) \cdot M_n.$$

The network robustness is improved at the expense of the signaling overhead. The trade-off between the robustness and overhead will be discussed for future works. In our paper, we focus on deriving the exact conditions of improving the network robustness by adding more layers of MEC servers.

V. Simulation Results

In this section, we evaluate the system latency and the processing rate in the HetMEC network performing our algorithm LMA and other task assignment schemes. The robustness in the HetMEC network with different number of layers is also investigated.

A. Parameters Setting

In our simulation, the parameters about the data processing are presented in Table II. The EDs transmit the file of size 60 Kbits in a period of 1s. We assume that one CPU cycle is required to process each bit of raw data, i.e., $\mu = 1$ cycles/bit. The compression ratio of the raw data after processing is set as 10%. The computing capacity is represented by the maximum CPU cycles per second, and the transmission resources are reflected by the transmission bandwidth. We consider four cases in our simulation, i.e., the cloud-only network, 1-layer, 2-layer and 3-layer HetMEC networks, as shown in Table III. In different cases, the network topology is fixed yet we consider the task assignment among different number of MEC

[15] This is to say, the right side of (31) becomes larger.

[16] This is to say, the left side of (31) becomes smaller.

TABLE III
NETWORK ARCHITECTURE OF THE HETMEC NETWORK WITH DIFFERENT NUMBER OF MEC LAYERS

	Cloud-only	One-layer HetMEC	Two-layer HetMEC	Three-layer HetMEC
Consideration of the EDs	✓	✓	✓	✓
Number of the MEC layer	0	1 (AP)	2 (AP, switch)	3 (AP, switch, network gateway)
Consideration of the CC	✓	✓	✓	✓

TABLE IV
COMPUTING CAPACITY AND TRANSMISSION RESOURCE SETTING OF THE HETMEC NETWORK

Node	Computing capacity (Mcps, Millon cycles per second)	Transmission resources (Mbps)
ED	0.12	-
AP	0.4	1.2
Switch (Lower-layer MEC server)	1.5	3
Network gateway (Upper-layer MEC server)	4.2	4.8
CC	12	12

layers. The computing capacity and transmission resource settings of the three cases are presented in Table IV. In all cases, the data generated at the EDs need to be aggregated at the CC through multiple layers of MEC servers, while the task assignment strategy can be performed among the EDs, all considered MEC layers and the CC.

B. Network Performance Evaluation

The network performance is evaluated based on the following metrics:

- **System Latency**: The total latency of all tasks generated at the EDs per second.
- **Processing Rate**: The average volume of data processed by the HetMEC network per second viewed by each ED.
- **Network Robustness**: The network robustness is represented by the maximum data generation speed supported by the non-congested HetMEC network.

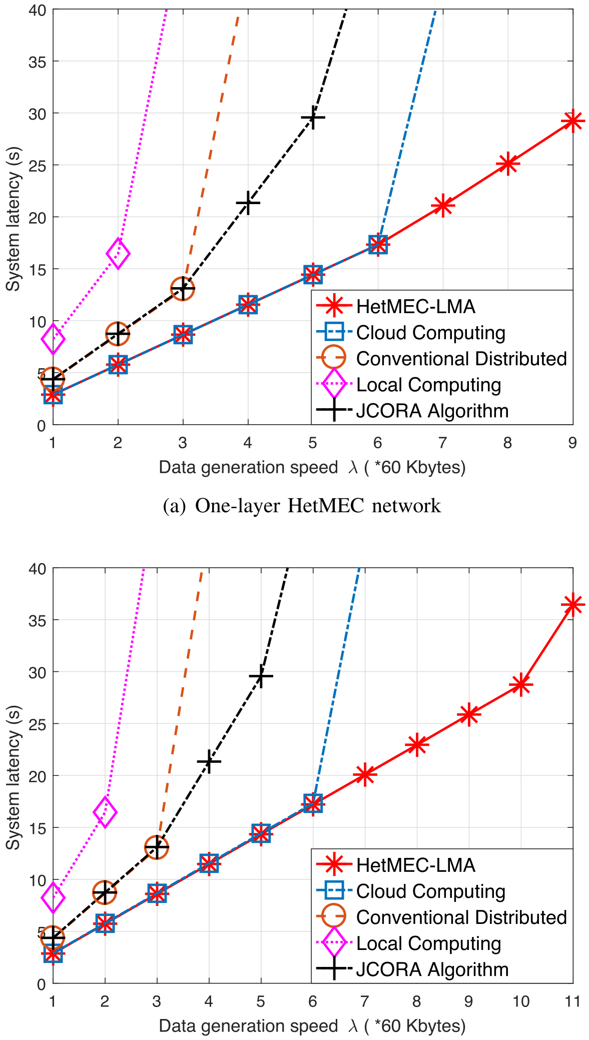
We compare the LMA with the following task assignment schemes on these metrics.

- *Cloud computing*: The cloud computing scheme indicates that all the data are processed at the CC.
- *Local computing*: The local computing scheme indicates that all the data are processed at the ED without being offloaded to the MEC servers or CC.
- *Conventional MEC*: The conventional MEC scheme indicates that the data are totally offloaded to the APs for processing, yet the CC, EDs and upper layer MEC servers do not provide the computing services.
- *JCORA Algorithm [29]*: The computation offloading between the APs as well as the EDs and the resource allocation scheme are adjusted iteratively in the joint computation offloading and resource allocation (JCORA) algorithm.

1) System Latency Evaluation: Fig. 6 presents the system latency versus the data generation speed λ at the ED in different cases. In the one-layer HetMEC network, as shown in Fig. 6(a), the system latency of all schemes increases with the data generation speed, since more data need to be processed, resulting in larger system latency. By performing our proposed algorithm LMA, the system latency remains the lowest given different data generation speed. When the data generation speed $\lambda > 6$, the slop of the line segment of LMA becomes larger, reflecting that the average latency increases. When the data generation speed is small, the latency is smallest when the data are processed at the uppermost layer. When the data generation speed is large, the data need to be offloaded to other layers due to the limited computing capacity of the uppermost layer, which induces the increase of the average latency.

In the two-layer HetMEC network, as shown in Fig. 6(b), the system latency of the LMA also remains the lowest in different cases given different data generation speeds, which reflects the advantages of the LMA in the HetMEC network. Compared with other schemes, the LMA jointly utilizes the computing capacity and transmission resources of all devices, and thus, its latency remains low with the data generation speed increasing. At the data generation speed $\lambda = 11$, the new generated data can be processed in real time when performing our algorithm LMA, showing the robustness of our scheme. Given the same data generation speed, the system latency of the two-layer HetMEC network performing the LMA is not larger than that of the one-layer HetMEC network, and the robustness of the two-layer HetMEC network performing the LMA is stronger than that of the one-layer HetMEC network.

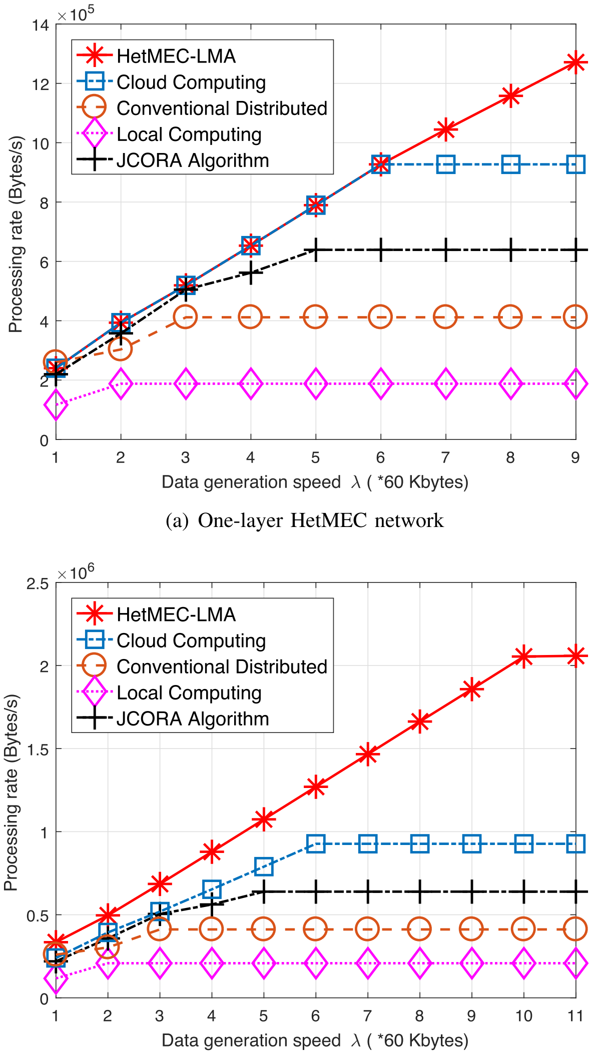
2) Processing Rate Evaluation: As shown in Fig. 7, we analyze the processing rate given different data generation speed in both cases. As presented in Fig. 7(a) and (b), both in the one-layer HetMEC network and two-layer HetMEC network, the processing rate of different schemes is non-decreasing as the data generation speed increases. Other schemes, e.g., JCORA algorithm, conventional MEC scheme, the cloud and local computing, reach the saturated point when



(a) One-layer HetMEC network

(b) Two-layer HetMEC network

Fig. 6. The system latency vs. the data generation speed in the different cases.



(a) One-layer HetMEC network

(b) Two-layer HetMEC network

Fig. 7. The processing rate with the increase of the data generation speed in the HetMEC network.

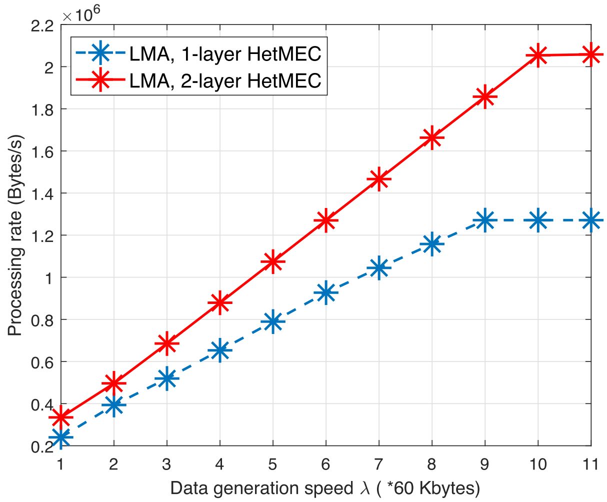


Fig. 8. The processing rate vs. the data generation speed in the HetMEC network with different number of layers.

the data generation speed surpasses the respective thresholds, reflecting the bottlenecks of the network processing rate utilizing these schemes. After reaching the bottleneck, the network cannot offer more computing resources for the new generated data, and the processing rate stops increasing. The LMA gains relatively high processing rate, especially when the data generation speed is large. Since the LMA jointly utilizes the computing and transmission resources of the whole HetMEC network, it achieves the highest bottleneck of the processing rate.

Fig. 8 compares the processing rate in the one-layer and two-layer HetMEC networks by performing LMA. When the HetMEC network is congested, no more resources can be utilized for data processing, and thus, the processing rate reaches saturation and stop grows with the data generation speed. Compared with the one-layer HetMEC network, the processing rate of the two-layer HetMEC network is higher given the same data generation speed, especially when the data generation speed is large. The computing and transmission resources are enriched in the two-layer HetMEC network. By jointly utilizing the computing resources of different layers and properly scheduling the data transmission among layers in the HetMEC network, more computing resources contributes to higher processing rate as the number of layers grows.

TABLE V

COMPUTING CAPACITY AND TRANSMISSION RESOURCE SETTING IN DIFFERENT CASES

Node	Case 1		Case 2		Case 3	
	Compute (Mcps)	Transmit (Mbps)	Compute (Mcps)	Transmit (Mbps)	Compute (Mcps)	Transmit (Mbps)
ED	0.12	-	0.12	-	0.12	-
AP	0.4	0.9	0.4	1.2	0.4	3
Switch	1.5	3	1.5	3	1.5	6
Network gateway	4.2	4.8	4.2	4.8	4.2	12
CC	12	12	12	12	12	15

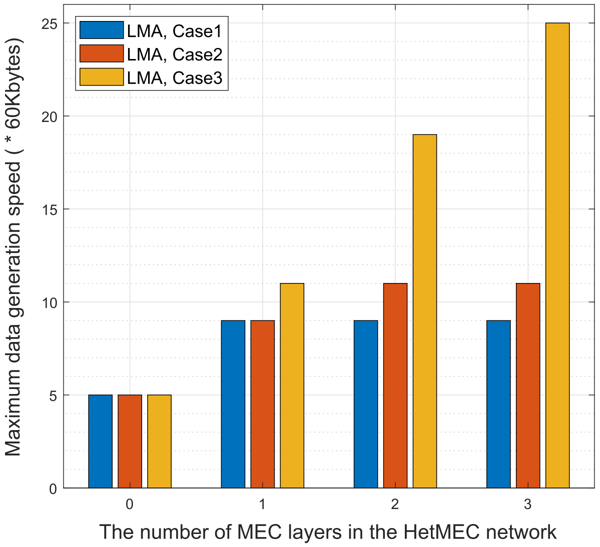


Fig. 9. The network robustness with different number of MEC layers in the HetMEC network.

3) Network Robustness Evaluation: Fig. 9 shows the maximum data generation speed of the HetMEC network in the cloud-only network, one-layer, two-layer and three-layer HetMEC networks in different cases, the settings of which are presented in Table V. The maximum data generation speed of the non-congested HetMEC network reflects the robustness of the network, as analyzed in Section IV. Case 1 and case 2 show the performance of robustness in the transmission resource shortage case of the HetMEC network, and case 3 shows the performance of robustness in the computing resource shortage case of the HetMEC network.

In case 1, as the number of the MEC layers increases, the network robustness first becomes stronger when the number of MEC layers $N \leq 1$, and then remains the same when $N \geq 1$. *When* $N \leq 1$, the main constraint of the robustness is the computing resources in the whole network. Therefore, the robustness of the HetMEC network become stronger when inducing more layers of MEC servers into the task assignment and processing. *When* $N \geq 1$, it is the transmission resources between the APs and EDs that constrains the network robustness. As analyzed in Section IV, it cannot contribute to the robustness enhancement to add a layer of MEC servers above the APs, and thus, the network robustness does not increase anymore.

In case 2, as the number of MEC layers grows, the network robustness becomes stronger when the number of MEC layers $N \leq 2$, and then remains the same when $N \geq 2$. Since the transmission resources between the APs and the EDs become more abundant, the HetMEC network can process the tasks with larger data generation speed, and thus, the robustness is enhanced in the two-layer HetMEC network. However, the transmission resources between the APs and the EDs still constrain the enhancement of the robustness, and the robustness remains unchanged when $N \geq 2$. The network robustness is improved only when the computing and transmission resources of the added layer satisfy the conditions analyzed in Section IV.

In case 3, the network robustness becomes stronger as the number of MEC layers grows. The transmission resources of all MEC layers are abundant, and the computing resources constrain the improvement of the robustness. As analyzed in Section IV, in the computing resource shortage case, it can improve the network robustness to induce new layers of MEC servers satisfying (25) and (26).

VI. CONCLUSION

In this paper, we have studied a HetMEC network in order to provide low-latency data services. We have considered a typical uplink MEC application, where the raw data are generated at the EDs and the results of the data processing need to be aggregated at the CC through multiple layers of MEC servers. The tasks are optimally divided and assigned to the nodes on multiple layers, including the CC, MEC servers and EDs. Through jointly considering the task assignment, computing and transmission resource allocation, we have proposed the LMA for latency minimization in the HetMEC network. Simulation results have showed that our proposed algorithm LMA can significantly reduce the system latency and increase the processing rate as well as the network robustness. Based on both theoretical and numerical analysis, we conclude that the relation between the network robustness and the number of layers of the HetMEC network is influenced by the amount of the computing and transmission resources. In the computing resource shortage case, the robustness can be improved when inducing the MEC servers above any layer. In contract, in the transmission resource shortage case, it cannot contribute to the enhancement of the robustness when inducing more layers of MEC servers above the initial transmission resource constrained layer.

APPENDIX A
PROOF OF PROPOSITION 2

In the proportional optimization problem (19), the task assignment strategy **s**, computing capacity allocation $\boldsymbol{\theta}$ and transmission resources allocation $\boldsymbol{\phi}$ are coupled, expressed as

$$L(\boldsymbol{s},\boldsymbol{\theta},\boldsymbol{\phi}) = \frac{\lambda_0^1}{\theta_0^1} + \sum_{n=1}^{N+1}\sum_{j=1}^{M_{n-1}}\sum_{i\in\mathcal{Q}_{n-1}^j}\left[\frac{s_n^i\lambda_n^i}{\theta_n^i} + \frac{\rho s_n^i\lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i}{\phi_n^{j,i}}\right] \tag{34}$$

However, it is worth noting that the transmission resources of each node that allocated to its child nodes are limited, as described in (13), and the upperbound of the computing capacity of each node is fixed. We consider that no spare computing capacity or transmission resource is left, i.e., equality occurs for (2) and (10). Hence, by utilizing the Cauchy-Schwarz inequality [34], we can obtain the following inequation.

$$\begin{aligned} &L(\boldsymbol{s},\boldsymbol{\theta},\boldsymbol{\phi}) \\ &\quad\geq L_{min}(\boldsymbol{s}) \\ &\quad= \left[\frac{\lambda_0^1}{\theta_0^{1,u}} + \sum_{n=1}^{N+1}\sum_{j=1}^{M_{n-1}}\sum_{i\in\mathcal{Q}_{n-1}^j}\frac{s_n^i\lambda_n^i}{\theta_n^{i,u}}\right] \\ &\qquad + \sum_{n=1}^{N+1}\sum_{j=1}^{M_{n-1}}\frac{\left(\sum_{i\in\mathcal{Q}_{n-1}^j}\sqrt{\rho s_n^i\lambda_n^i + (1-s_n^i)\lambda_n^i + \beta_n^i}\right)^2}{\phi_{n-1}^j}, \end{aligned} \tag{35}$$

where $\theta_n^{i,u}$ and ϕ_{n-1}^j are the boundary of the computing and transmitting capacity. The proportional optimization problem (19) is converted into a pure task assignment problem.

Once the equation conditions of the Cauchy-Schwarz inequality are satisfied, the task assignment problem is completely equivalent to the latency minimization problem $\min L(\boldsymbol{s},\boldsymbol{\theta},\boldsymbol{\phi})$. Therefore, we can obtain the optimal solution of $\min L(\boldsymbol{s},\boldsymbol{\theta},\boldsymbol{\phi})$ by solving $\min L_{min}(\boldsymbol{s})$.

APPENDIX B
PROOF OF THE PROPOSITION 4

We analyze the network with one parent node and M child nodes. Let s_i and λ_i denotes the task assignment percentage and raw data arriving rate at child node i. The maximum computing capacity of child node i is denoted by θ_i^u, and the computing capacity of the parent node is denoted by θ^u. The total transmission resource of the parent node is expressed by ϕ.

The latency L_{min} can be expressed by

$$L_{min} = \sum_{i=1}^{M}\frac{s_i\lambda_i}{\theta_i^u} + \frac{\sum_{i=1}^M(1-s_i)\lambda_i}{\theta^u} + \frac{\left(\sum_{i=1}^M\sqrt{(1-s_i)\lambda_i+\rho s_i\lambda_i}\right)^2}{\phi}.$$

The Hessian matrix of the system latency with M child nodes can be expressed by

$$\mathbf{H}_M = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,M} \\ h_{2,1} & h_{2,2} & \dots & h_{2,M} \\ \dots & \dots & & \dots \\ h_{M,1} & h_{M,2} & \dots & h_{M,M} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2\ L_{min}}{\partial s_1\ s_1} & \frac{\partial^2\ L_{min}}{\partial s_1\ s_2} & \dots & \frac{\partial^2\ L_{min}}{\partial s_1\ s_M} \\ \frac{\partial^2\ L_{min}}{\partial s_2\ s_1} & \frac{\partial^2\ L_{min}}{\partial s_2\ s_2} & \dots & \frac{\partial^2\ L_{min}}{\partial s_2\ s_M} \\ \dots & \dots & & \dots \\ \frac{\partial^2\ L_{min}}{\partial s_M s_1} & \frac{\partial^2\ L_{min}}{\partial s_M s_2} & \dots & \frac{\partial^2\ L_{min}}{\partial s_M s_M} \end{bmatrix}$$

We can obtain the second partial derivative as follows

$$h_{i,i} = \frac{\partial^2 L_{min}}{\partial s_i^2} = -Z\lambda_i^2\frac{(\sum_{j=1}^M A_j) - A_i}{A_i^3}, \tag{36}$$

$$h_{i,j} = \frac{\partial^2 L_{min}}{\partial s_i s_j} = Z\lambda_i\lambda_j\frac{1}{A_iA_j}, \tag{37}$$

where

$$Z = \frac{(1-\rho)^2}{2\phi} \geq 0, \tag{38}$$

$$A_i = \sqrt{(1-s_i)\lambda_i + \rho s_i\lambda_i} \geq 0. \tag{39}$$

Considering a normal vector $\mathbf{x} = [x_1\ x_2 \dots x_M]^T$, we obtain the following polynomial

$$X_M(\mathbf{x}) = \mathbf{x}^T\mathbf{H}_M\mathbf{x} = \sum_{i=1}^M\sum_{j=1}^M h_{i,j}x_ix_j. \tag{40}$$

We then prove that $X_M(\mathbf{x}) \leq 0$ for any natural number M by mathematical induction.

- When $M = 1$: $X_1(\mathbf{x}) = 0 \leq 0$
- When $M = 2$: $X_2(\mathbf{x}) = -Z\frac{(A_1^2\lambda_2 - A_2^2\lambda_2)^2}{M_1^3\ M_2^3} \leq 0.$
- We assume that when $M = m-1$, $X_{m-1}(\mathbf{x}) = \mathbf{x}^T\mathbf{H}_{m-1}\mathbf{x} \leq 0.$
- Hence, when $M = m$, we have

$$\begin{aligned} X_m(\mathbf{x}) &= \mathbf{x}^T\mathbf{H}_m\mathbf{x} = \sum_{i=1}^m\sum_{j=1}^m h_{i,j}x_ix_j \\ &= X_{m-1} - \frac{Z}{A_m^3}\sum_{i-1}^{m-1}\frac{(A_m^2\lambda_i - A_i^2\lambda_m)^2}{A_i^3} \leq 0. \end{aligned}$$

Since $X_M(\mathbf{x}) \leq 0$ for any natural number M, the Hessian matrix H_M is a seminegative definite matrix, implying that the function L_{min} is concave [35].

Moreover, the non-congested constraints are linear based on the **Proposition 1**. Hence, the minimum value of a concave function is obtained at the vertex of the feasible set bounded by the non-congested constraints.

APPENDIX C
PROOF OF REMARK 2

We consider the worst case, that the number of child nodes connected with each parent node is $Q = \max_{0\leq n\leq N, 1\leq i\leq M_n} Q_n^i$. In this case, the number of nodes in the whole network can be calculated as below.

$$M = \sum_{n=0}^{N+1} Q^n = \frac{Q^{N+2}-1}{Q-1}. \tag{41}$$

The complexity of the latency minimization algorithm is proportional to the number of feasible vertices, which is proportional to the square of the number of the constraints and closely related to the complexity of finding the vertices. The non-congested constraints derive from the computing capacity limitation of each node and the transmission resources limitation of each parent node. The number of the computing capacity constraints equals that of all nodes:

$$K_c = M = \sum_{n=0}^{N+1} Q^n = \frac{Q^{N+2}-1}{Q-1}. \tag{42}$$

The number of the transmission resource constraints equals the number of the parent nodes in the whole network, which can be expressed by

$$K_t = \sum_{n=0}^{N} Q^n = \frac{Q^{N+1}-1}{Q-1}. \tag{43}$$

The number of the constraints is

$$K = K_c + K_t = \frac{Q^{N+2}+Q^{N+1}-2}{Q-1}. \tag{44}$$

The maximum number of the vertices of the feasible set is

$$O(K^2) = \frac{O(Q^{2N+4})}{O(Q^2)} = O(M^2). \tag{45}$$

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of Things: Vision, applications and research challenges,” *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[3] D. Evans, “The Internet of Things: How the next evolution of the Internet is changing everything,” CISCO, San Jose, CA, USA, White Paper, Jan. 2011, vol. 1, pp. 1–11.

[4] A. Papageorgiou, B. Cheng, and E. Kovacs, “Real-time data reduction at the network edge of Internet-of-Things systems,” in *Proc. CNSM*, Barcelona, Spain, Nov. 2015, pp. 284–291.

[5] M. Armbrust *et al.*, “A view of cloud computing,” *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—A key technology towards 5G,” ETSI, Sophia Antipolis, France, White Paper 11, Sep. 2015.

[7] P. Wang, B. Di, H. Zhang, K. Bian, and L. Song, “Cellular V2X communications in unlicensed spectrum: Harmonious ooexistence with VANET in 5G systems,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5212–5224, Aug. 2018.

[8] 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.

[9] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: Architecture, applications, and approaches,” *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Oct. 2011.

[10] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[11] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[13] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A survey on mobile edge networks: Convergence of computing, caching and communications,” *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[14] L. Yang, J. Cao, H. Cheng, and Y. Ji, “Multi-user computation partitioning for latency sensitive mobile cloud applications,” *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.

[15] 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.

[16] 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.

[17] *Network Architecture*, document 3GPP TS 23.002, Jun. 2003.

[18] A. S. Tanenbaum and D. Wetherall, *Computer Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[19] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for Internet of Things and analytics,” in *Big Data and Internet of Things: A Roadmap for Smart Environments*, vol. 546. Cham, Switzerland: Springer, Mar. 2014, pp. 169–186.

[20] L. Tong, Y. Li, and W. Gao, “A hierarchical edge cloud architecture for mobile computing,” in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[21] A. Kiani and N. Ansari, “Toward hierarchical mobile edge computing: An auction-based profit maximization approach,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.

[22] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen, “Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach,” *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep. 2017.

[23] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, “Energy-efficient fault-tolerant data storage and processing in mobile cloud,” *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, Jul. 2015.

[24] 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.

[25] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, “Computation offloading and resource allocation in wireless cellular networks with mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[26] S. Ko, K. Han, and K. Huang, “Wireless networks for mobile edge computing: Spatial modeling and latency analysis,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5225–5240, Aug. 2018.

[27] 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.

[28] T. X. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2018. doi: 10.1109/TVT.2018.2881191.

[29] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[30] H. Dahrouj, A. Douik, F. Rayal, T. Y. Al-Naffouri, and M.-S. Alouini, “Cost-effective hybrid RF/FSO backhaul solution for next generation wireless systems,” *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 98–104, Oct. 2015.

[31] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[32] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[33] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[34] J. Michael Steele, *The Cauchy–Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Pengfei Wang (S'18) received the B.S. degree in electronics engineering from Peking University, Beijing, China, in 2017, where he is currently pursuing the master's degree with the Department of Electronics. His current research interests include wireless communications, vehicular networks, and edge computing.

Zijie Zheng (S'14) received the Ph.D. degree in signal processing from the Department of Electronics, Peking University, China, in 2019. His current research interests include game theory and optimization in 5G networks, wireless powered networks, mobile social networks, and wireless big data.

Boya Di (S'17–M'19) received the B.S. degree in electronic engineering from Peking University, China, in 2014, and the Ph.D. degree from the Department of Electronics, Peking University in 2019. Her current research interests include edge computing, vehicular networks, non-orthogonal multiple access, and 5G wireless networks. So far she has contributed as the first author for nine journal articles and one of her journal papers is currently listed as ESI highly cited papers. She has also served as a TPC member in GlobeCom 2016, ICCC 2017, ICC 2016, ICC 2018, and VTC 2019.

Lingyang Song (S'03–M'06–SM'12–F'19) received the Ph.D. degree from the University of York, U.K., in 2007, where he received the K. M. Stott Prize for excellent research. He was a Research Fellow with the University of Oslo, Norway, until rejoining Philips Research U.K. in March 2008. In May 2009, he joined the Department of Electronics, School of Electronics Engineering and Computer Science, Peking University, and is currently a Boya Distinguished Professor. His main research interests include wireless communication and networks, signal processing, and machine learning. He was a recipient of the IEEE Leonard G. Abraham Prize in 2016 and the IEEE Asia–Pacific (AP) Young Researcher Award in 2012. He has been an IEEE Distinguished Lecturer since 2015.