Distributed Energy Management for Multiuser Mobile-Edge Computing Systems With Energy Harvesting Devices and QoS Constraints

Guanglin Zhang, *Member, IEEE*, Yan Chen, Zhirong Shen, and Lin Wang, *Member, IEEE*

***Abstract*—Mobile-edge computing (MEC) has evolved as a promising technology to alleviate the computing pressure of mobile devices by offloading computation tasks to MEC server. Energy management is challenging since the unpredictability of the energy harvesting (EH) and the quality of service (QoS). In this paper, we investigate the problem of power consumption in a multiuser MEC system with EH devices. The system power consumption, which includes the local execution power and the offloading transmission power, is designated as the main system performance index. First, we formulate the power consumption minimization problem with the battery queue stability and QoS constraints as a stochastic optimization programming, which is difficult to solve due to the time-coupling constraints. Then, we adopt the Lyapunov optimization approach to tackle the problem by reformulating it into a problem with relaxed queue stability constraints. We design an online algorithm based on the Lyapunov optimization method, which only uses current states of the mobile users and does not depend on the system statistic information. Furthermore, we propose a distributed algorithm based on the alternating direction method of multipliers to reduce the system computational complexity. We prove the optimality of the online algorithm and the distributed algorithm using rigorous theoretical analysis. Finally, we perform extensive trace-simulations to verify the theoretical results and evaluate the effectiveness of the proposed algorithms.**

***Index Terms*—Alternating direction method of multipliers (ADMM), energy harvesting (EH), Lyapunov optimization, mobile-edge computing (MEC), quality of service (QoS).**

I. Introduction

WITH the rapid development of the Internet of Things, mobile devices are in an exploding speed to occupy each aspect of our lives, such as online trading, entertainment, daily learning, etc., [1], [2]. However, mobile devices are faced with a challenge that it is difficult to compute the massive and intensive tasks quickly and timely with the limited computing capability and resource [3]. Therefore, it is emerging to seek out new techniques to resolve the impasse. Recently, mobile-edge computing (MEC) is becoming a promising technique to alleviate the computation stress of mobile users (MUs) [4], [5]. MEC technology combines the wireless network and the Internet technology effectively, and increases the computing, the storage, the processing and other functions on the wireless network side [6]. In MEC systems, MUs can offload all or partial computation tasks to MEC server in order to improve the computation speed and quality [5]. It is similar to construct an open platform to reduce network latency for MUs by providing resources and computing capability at the edge network [7].

It has been an urgent need to increase the energy efficiency of mobile devices with the increasing computing energy consumption and the battery problem. Energy harvesting (EH) technology is emerging as a promising paradigm to solve the above problems and make the green computing communication to come true [8]. EH devices can harvest energy from the environment, such as the solar power, the wind energy, and human kinetic energy [9], [10]. In recent work [11], it investigated the tradeoff between the power consumption and the execution delay in a multiuser MEC system. In [12], it studied resource management and allocation for EH cognitive radio sensor networks. Compared to the recent works, the investigation of energy management for multiuser MEC system with EH devices remains unknown.

In addition to energy management problem, quantifying the users' experience is of great importance in MEC systems as well. In [13], it proposed a proactive service framework to minimize the delay while bounding the time-average energy consumption. In daily use, the backlog of data and the delay of execution are the two major factors that affect the MUs' experience. To guarantee the quality of service (QoS) of MUs and the stability of the system, we put the focus on the balance between the data arrival and execution. In this paper, we define a probability to represent the proportion of data that is not processed in time to total data arrival as a decision of QoS [14]. The probability will be kept below a target value to guarantee the QoS constraint under the proposed algorithms.

In this paper, we consider a multiuser MEC system with EH devices. The execution model consists of the local execution and the MEC server execution. In particular, we add

Manuscript received May 14, 2018; revised July 20, 2018; accepted October 6, 2018. Date of publication October 15, 2018; date of current version June 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61772130, Grant 61873176, Grant 61301118, and Grant 61473189, in part by the International S&T Cooperation Program of Shanghai Science and Technology Commission under Grant 15220710600, in part by the Natural Science Foundation of Shanghai under Grant 17ZR1445200, and in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 14YZ130. *(Corresponding author: Guanglin Zhang.)*

G. Zhang, Y. Chen, and Z. Shen are with the College of Information Science and Technology, Donghua University, Shanghai 201620, China, and also with the Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, Donghua University, Shanghai 201620, China (e-mail: glzhang@dhu.edu.cn; yanchen@mail.dhu.edu.cn; szrong@mail.dhu.edu.cn).

L. Wang is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wanglin@sjtu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2018.2875909

a sleep-awake decision into the system to improve the effective utilization of the harvested energy. The objective is to minimize the power consumption of the system. The formulated power minimization problem subjects to the operational constraints and the heavy computation due to the centralized control of multiuser. To overcome the challenges, we develop the Lyapunov optimization-based online algorithm [15] and the distributed algorithm based on alternating direction method of multipliers (ADMM) method [16].

The major contributions are summarized as follows.

1) We consider a multiuser MEC system with EH devices to investigate the energy management problem. The objective is to minimize the power consumption of the system and ensure the users' QoS. It is challenging due to the stochastic commodity arrivals, the intermittent EH, and the bandwidth allocation.
2) We formulate the power consumption minimization problem as a stochastic optimization programming with the QoS and the battery time-coupling constraints. A Lyapunov-based centralized energy management algorithm (LYP-CEMA) is proposed to minimize the power consumption and decouple the time-coupling of QoS and battery virtual queues. We propose the online algorithm that only depends on the current states of system without future knowledge. Moreover, we develop an ADMM-based distributed energy management algorithm (ADMM-DEMA) based on LYP-CEMA. Under the distributed algorithm, each user can make decisions on their own and achieved a lower computational complexity compared with the centralized algorithm.
3) Performance analysis reveals the asymptotic optimality of the Lyapunov-based optimization algorithm by selecting an appropriate value of control parameter V. Besides, we prove the upper bound of the two virtual queues and guarantee the feasibility of the proposed algorithm. Simulation results verify the theoretical analysis and show the efficiency on power consumption minimization and optimality of the proposed algorithms LYP-CEMA and ADMM-DEMA.

This paper is organized as follows. In Section II, we introduce the system model. In Section III, we formulate the power consumption minimization problem. In Section IV, we develop an LYP-CEMA and discuss the theoretical performance analysis of the algorithm. In Section V, we propose an ADMM-DEMA. In Section VI, the empirical evaluation is presented and discussed. In Section VII, we introduce the related work. Finally, we conclude this paper in Section VIII.

II. System Model

We consider a multiuser MEC system with EH devices as shown in Fig. 1. MUs can execute partial commodities at the local CPU and offload some intensive commodities to the MEC server as shown in Fig. 2. The MEC server can be accessed to MUs through the wireless channels and can execute the offloaded computation commodities at the edge network. With the powerful computing capability of the MEC server, the quality of computation experience in each MU can be improved greatly [17].

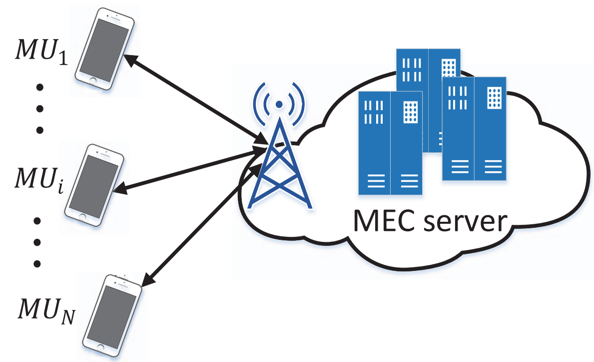


Fig. 1. MEC system with N MUs.

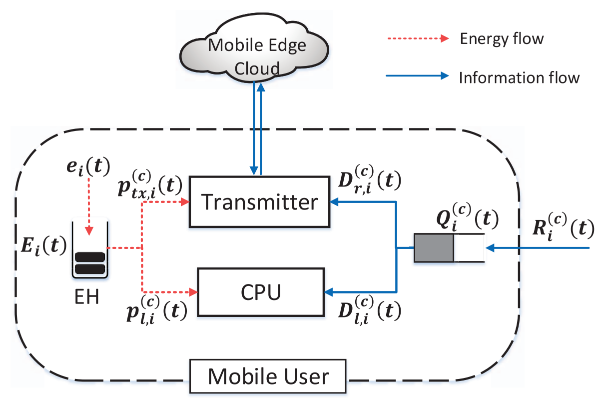


Fig. 2. System architecture of one MU.

Besides, each MU is equipped with an EH device as shown in Fig. 2. EH devices are capable of harvesting energy from environment, such as the solar power, the wind energy, and human kinetic energy. Each MU is powered only by the harvested renewable energy from the EH device. We denote the harvested energy arrivals for MU i at time t as $\varepsilon_i(t)$ and assume that $\varepsilon_i(t)$ are independent and identically distributed (i.i.d.) in different time slots with the maximum value $\varepsilon_i^{\max}$. Actually, only a part of the harvested energy arrivals can be stored into the battery. To model the harvesting decision, we use $e_i(t) \in [0, \varepsilon_i(t)]$ to denote the harvesting energy charged to battery for MU i at time t. The energy flow is described in Fig. 2.

Moreover, to reduce the energy consumption and improve the utilization efficiency of the harvested renewable energy, we add the sleep-awake decision to each MU, i.e., each MU can choose to stay awake or enter the state of sleep at the beginning of each time slot. We use $1_i(t)$ to denote the sleep-awake decision and define it as follows:

$$1_i(t) = \begin{cases} 1, & \text{if the MU } i \text{ stays awake at time slot } t \\ 0, & \text{otherwise.} \end{cases}$$

which means that if there exists data to process, then the decision will be set as 1. And if no data arrive in time slot t, the decision will be set as 0.

We assume that time is slotted and denote the time slot length as τ. For convenience, we use $\mathcal{N} \triangleq \{1, \ldots, N\}$ and

TABLE I
KEY NOTATIONS

Notation	Definition
$\mathcal{T}$	Index set of time slots
$\mathcal{N}$	Index set of mobile users (MUs)
$\mathcal{C}$	Index set of commodity flows
$\mathcal{A}$	Index set of the bandwidth allocation vector $\{\boldsymbol{\alpha}(t)\}$
$\alpha_i^{(c)}(t)$	Proportion of bandwidth allocated for commodity c of MU i at time t
$f_i^{(c)}(t)$	CPU-cycle frequency for commodity c of MU i at time t
$f_i^{(c),\max}$	Maximum allowable CPU-cycle frequency for commodity c of MU i
$D_{l,i}^{(c)}(t)$	Data length for commodity c executed at the local CPU of MU i at time t
$D_{r,i}^{(c)}(t)$	Data length for commodity c of MU i offloaded to the MEC server at time t
$R_i^{(c)}(t)$	New data processing requests admitted into commodity c of MU i at time t
$Q_i^{(c)}(t)$	Backlog of data queue for commodity c of MU i at time t
$\delta_i^{(c)}$	The probability of data that cannot be processed in time to total data for commodity c of MU i
$p_{l,i}^{(c)}(t)$	Power consumption in local execution for commodity c of MU i at time t
$p_{tx,i}^{(c)}(t)$	Power consumption in MEC transmission for commodity c of MU i commodity c at time t
$E_i(t)$	Battery energy level at time slot t for MU i
$E_i^{\max}$	Capacity of $E_i(t)$ for MU i
$\varepsilon_i(t)$	Harvesting energy arrivals for MU i at time t
$e_i(t)$	Harvesting energy charged to battery for MU i at time t

$\mathcal{T} \triangleq \{0, 1, \ldots, T\}$ to denote the index sets of MUs and the time slots, respectively. For ease of reference, we list the key notations of our system model in Table I. In the following, we introduce the details of each component of the system.

A. Execution Model

Each MU can provide computing and communication services to its running applications. We use $\mathcal{C} \triangleq \{1, \ldots, C\}$ to denote the set of the commodity flows generated by the running applications of MU i. We use $R_i^{(c)}(t)$ to denote the new data processing requests admitted into commodity c of MU i at time t and the new commodity data requests are accepted continuously by MU i in each time slot t. We assume that $R_i^{(c)}(t)$ is i.i.d. over different time slots without loss of generality and satisfies $R_i^{(c),\min} \le R_i^{(c)}(t) \le R_i^{(c),\max}$. Note that in practical scenarios, since the data arrival is random, which means that at some time slots, the arrival of the data may be zero. Thus, according to the definition of the sleep-awake decision, we have that if there is no data arrival to be processed, then the sleep-awake decision will be set as 0, i.e., the MU will be in the sleep mode.

In the following sections, we introduce the local execution model and the MEC server execution model in detail. The information flow is described in Fig. 2.

1) Local Execution Model: We denote the part of the computation data that will be executed at the local CPU of the MU i at time slot t as $D_{l,i}^{(c)}(t)$. We assume that in the local execution model, L_i CPU cycles will be needed for computing one bit admitted commodities. We use $f_i^{(c)}(t)$ to denote the scheduled CPU-cycle frequency for commodity c of MU i at time t, and the maximum allowable CPU-cycle frequency is defined as $f_i^{(c),\max}$, i.e.,

$$0 \le f_i^{(c)}(t) \le f_i^{(c),\max}, i \in \mathcal{N}, t \in \mathcal{T}. \tag{1}$$

Therefore, the data length for commodity c executed at the local CPU of MU i at time t, $D_{l,i}^{(c)}(t)$, can be expressed as follows:

$$D_{l,i}^{(c)}(t) = \tau f_i^{(c)}(t) L_i^{-1}. \tag{2}$$

In local execution model, the main energy consumption is generated by the CPU operation and the energy consumption is proportional to the square of the frequency of mobile device. We denote the power consumption for local execution for commodity c of MU i at time t as $p_{l,i}^{(c)}(t)$, given as follows:

$$p_{l,i}^{(c)}(t) = \kappa \left[f_i^{(c)}(t) \right]^2 \tag{3}$$

where κ is the effective switched capacitance related to the chip architecture [18].

2) MEC Server Execution Model: In the MEC server execution model, the computation data need to be transmitted to the MEC server which accessed to the MUs. We assume that the computation latency of MEC server is negligible due to the powerful computing capability and the great resources of MEC server. The available system bandwidth is w Hz, which is shared by the commodities of MUs, and the noise power spectral density at the receiver is denoted as N_0. And the channel power gain from MU i to the MEC server is denoted as H.

Let $p_{tx,i}^{(c)}(t)$ denote the data transmission power consumption for commodity c of MU i with the maximum value of $p_{tx}^{(c),\max}$, i.e.,

$$0 \le p_{tx,i}^{(c)}(t) \le p_{tx,i}^{(c),\max}, i \in \mathcal{N}, t \in \mathcal{T}. \tag{4}$$

Hence, the data length offloaded to the MEC server for commodity c of MU i at time t is shown by

$$D_{r,i}^{(c)}(t) = \begin{cases} \alpha_i^{(c)}(t) w \tau \log_2\left(1 + \frac{H p_{tx,i}^{(c)}(t)}{N_0}\right), & \alpha_i^{(c)}(t) > 0 \\ 0, & \alpha_i^{(c)}(t) = 0 \end{cases} \tag{5}$$

where $\alpha_i^{(c)}(t)$ is the proportion of bandwidth allocated for commodity c of MU i at time t. The bandwidth allocation vector $\vec{\boldsymbol{\alpha}}(t) = [\alpha_1^{(c)}(t), \ldots, \alpha_N^{(c)}(t)]$ should be chosen from the feasible set $\mathcal{A}$ [19] as follows:

$$\vec{\boldsymbol{\alpha}}(t) \in \mathcal{A} \triangleq \left\{ \boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{NC} | \sum_{i \in \mathcal{N}} \sum_{c \in \mathcal{C}} \alpha_i^{(c)}(t) \leq 1 \right\}. \tag{6}$$

3) Execution QoS Model: The backlog of data and the execution delay are the two major factors that affect the MUs' experience in daily use. We assume that the computation latency of MEC server is negligible due to the powerful computing capability and the great resources of MEC server. Then, to guarantee the QoS and the stability of the system, we put the focus on the balance between the data arrival and execution in this part.

First, we assume that each commodity of MU i can tolerate a prescribed probability, $\delta_i^{(c)}$, to represent the proportion of data that cannot be processed in time to total data arrival for commodity c of MU i. Note that $\delta_i^{(c)}$ is varied for different commodities of MUs. That means a part of the data arrival will be queued and executed at the next slot. We denote the backlog of the data queue for commodity c of MU i as $Q_i^{(c)}(t)$ at time slot t. It is updated by

$$Q_i^{(c)}(t+1) = Q_i^{(c)}(t) + R_i^{(c)}(t) - 1_i(t)\Big(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)\Big). \tag{7}$$

Let $\mathbf{Q}(t) \triangleq [Q_1^{(c)}(t), Q_2^{(c)}(t), \ldots, Q_N^{(c)}(t)]$ denote the set of the data backlogs for all i and c.

Besides, since the data request of each time slot include the data backlog of last time slot $t-1$ and the data arrival of current time slot t, we denote the average data request rate for commodity c of MU i as $\lambda_i^{(c)}$, i.e., $\lambda_i^{(c)} = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}(Q_i^{(c)}(t)+R_i^{(c)}(t))$. Meanwhile, let $\rho_i^{(c)}$ be the average data backlog rate for commodity c of MU i, i.e., $\rho_i^{(c)} = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1} Q_i^{(c)}(t+1)$. Thus, according to the prescribed probability of data that is not completed in time, $\delta_i^{(c)}$, we have

$$\rho_i^{(c)} \leq \delta_i^{(c)} \cdot \lambda_i^{(c)} \tag{8}$$

which means that the data backlog should always be up bounded depending on the prescribed probability to guarantee the QoS. Note that the constraint should be satisfied in the long time and the not completed data request of time slot t can be queued and executed in the next slot under the QoS constraint. Later, we will transform the QoS constraint to a virtual queue stability constraint and prove the uniformity.

B. Battery Dynamic Model

Each MU is equipped with an EH device and is powered only by $e_i(t)$, the actual harvested energy charged to battery for MU i at time t. The battery energy level of MU i at the beginning of time slot t is denoted as $E_i(t)$. Consider that the capacity of the battery for storing the harvested energy is finite. The battery energy level $E_i(t)$ is bounded as follows:

$$E_i^{\min} \leq E_i(t) \leq E_i^{\max} \tag{9}$$

where $E_i^{\min}$ is the minimum allowed energy level, and $E_i^{\max}$ is the maximum allowed energy level.

The total system power consumption consists of the power consumption in local execution for commodity c of MU i, $p_{l,i}^{(c)}(t)$, and the power consumption of the transmission for computation offloading to the MEC server, $p_{tx,i}^{(c)}(t)$. Hence, the total system power consumption is denoted as $P(t)$

$$P(t) = \sum_{i\in\mathcal{N}} \sum_{c\in\mathcal{C}} 1_i(t)\Big(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big). \tag{10}$$

Therefore, the battery energy level evolves according to the following equation:

$$E_i(t+1) = E_i(t) - \sum_{c\in\mathcal{C}} 1_i(t)\Big[p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big] + e_i(t). \tag{11}$$

Since we only can utilize the actual harvested energy that has been stored in the battery, so the power consumption at each time t must satisfy the energy operational constraint

$$0 \leq \sum_{c\in\mathcal{C}} 1_i(t)\Big(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big) \leq E_i(t). \tag{12}$$

III. Problem Formulation

In this section, we formulate the power consumption minimization problem as a stochastic optimization programming. It is difficult to solve due to the time-coupling constraints of the QoS and the battery. A relaxed problem is shown after loosening the time-coupling constraints.

The total system power consumption, $P(t)$, consists of the local execution power consumption and the transmission power for computation offloading in MEC server. We denoted the system average power consumption as $\overline{P}$ which is designated as the main performance metric

$$\overline{P} = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} P(t)\right]. \tag{13}$$

Based on the previous system model description, we define that the all control actions at time slot t by $\vec{\mathbf{u}}(t) = [f_i^{(c)}(t), p_{tx,i}^{(c)}(t), \alpha_i^{(c)}(t), e_i(t), 1_i(t)], \forall i, c$. Then, we formulate the power consumption optimization problem as a stochastic optimization problem as follows:

$$\begin{aligned} \text{P1}: \ & \min_{\{\vec{\mathbf{u}}(t)\}} \ \overline{P} \\ & \text{s.t.} \ (1), (4), (6), (8), (9), (11), (12) \end{aligned}$$

where (1), (4) are the feasible scope of the CPU-cycle frequency and the transmission power consumption, respectively. Equation (6) means the bandwidth allocation constraint and (8) means the QoS constraint. Equations (9) and (11) are the energy level bound and the energy queue stability constraints. Constraint (12) means the power consumption can not exceed the operational upper bound.

P1 is a stochastic programming problem since that the EH and the new data arrivals are random. Specifically, the battery energy level and data queue length of current time slot all depend on the previous time slot. It is challenging to solve the problem with the time-coupling constraints.

Next, we relax the time-coupling constraints (9) and (11) to the time-averaged constraint as shown

$$\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1}E_i(t)\right]=0, i\in\mathcal{N}. \tag{14}$$

With the above relaxation, we transform P1 into a relaxed problem as shown below

$$\begin{aligned}\text{P2}: \ &\min_{\{\vec{\mathbf{u}}(t)\}} \ \overline{P}\\ &\text{s.t.} \ (1), (4), (6), (8), (12), (14).\end{aligned}$$

In P2, the battery constraints (9) and (11) are replaced by (14) compared with P1. Based on the above relaxation, Lyapunov optimization approach can be adopted to solve P2. Note that the feasible solutions to P2 may be infeasible to P1 since the constraints relaxed in P2 and the formulation of P2 is to design an online algorithm for P1 in the next section and we will show the asymptotic optimality in performance analysis.

IV. Online Energy Management Algorithm

In this section, we first adopt two virtual queues and then design an online energy management algorithm based on Lyapunov optimization approach [15]. We also show the properties and the performance analysis of the asymptotic optimal solution under the LYP-CEMA.

A. Lyapunov Optimization Algorithm

First, We adopt a battery virtual queue $\tilde{E}_i(t)$ [20] to track the energy level of each battery of MU i

$$\tilde{E}_i(t)=E_i(t)-\theta_i \tag{15}$$

where θ_i is a perturbation parameter, which will be defined later in Section IV-C. With carefully selecting the value of θ_i, ample energy will be available in the battery for the communication and computation of MU i when it is awake. Hence, we have the virtual battery queue evolves as follows:

$$\tilde{E}_i(t+1)=\tilde{E}_i(t)-1_i(t)\left[\sum_{c\in\mathcal{C}}\left(p_{l,i}^{(c)}(t)+p_{tx,i}^{(c)}(t)\right)\right]+e_i(t). \tag{16}$$

Then, we adopt a QoS virtual queue $Z_i^{(c)}(t)$ [14], and its dynamics is shown as follows:

$$\begin{aligned}Z_i^{(c)}(t+1)=&\left[Z_i^{(c)}(t)-\delta_i^{(c)}\cdot\left(Q_i^{(c)}(t)+R_i^{(c)}(t)\right)\right]^+\\ &+Q_i^{(c)}(t)+R_i^{(c)}(t)-1_i(t)\left[D_{r,i}^{(c)}(t)+D_{l,i}^{(c)}(t)\right]\end{aligned} \tag{17}$$

where $[x]^+=\max\{0,x\}$.

Theorem 1: If we make the QoS virtual queue $Z_i^{(c)}(t)$ be always stable under the proposed policy, the quality execution data residue of MU i will also be stabilized at the average QoS rate $\rho_i^{(c)}\leq\delta_i^{(c)}\cdot\lambda_i^{(c)}$.

Proof: See Appendix A. ■

Utilizing the Lyapunov Optimization, the time coupling of battery queue and QoS queue can be decoupled and decisions can be made online which only uses current states of MUs. Let $\overrightarrow{U}(t)=(Z_i^{(c)}(t),\tilde{E}_i(t))$ and we define the *Lyapunov function* for virtual queues as follows:

$$L(t)=\frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}{Z_i^{(c)}(t)}^2+\frac{1}{2}\sum_{i\in\mathcal{N}}\tilde{E}_i(t)^2. \tag{18}$$

Next, the one-step conditional *Lyapunov drift* is defined as

$$\Delta L(t)=\mathbb{E}\Big[L(t+1)-L(t)|\overrightarrow{U}(t)\Big]. \tag{19}$$

Then, we introduce a parameter V to make a tradeoff between the queue stability and the system power consumption. The Lyapunov *drift-plus-penalty* function can be written as

$$\Delta_V L(t)=\Delta L(t)+V\cdot\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big]. \tag{20}$$

where $0<V\leq V_{\max}=\min_i\{E_i^{\max}-p_{tx,i}^{(c),\max}-\kappa[f_i^{(c),\max}L_i^{-1}]^2-\varepsilon_i^{\max}\}$. The value of $V_{\max}$ will be selected carefully to ensure the battery constraints (9) always be satisfied. Next, we show the upper bound of the $\Delta_V L(t)$ under the control actions $\vec{\mathbf{u}}(t)$ in Lemma 1.

Lemma 1: For any feasible $\vec{\mathbf{u}}(t)$, $\Delta_V L(t)$ has a upper bound as follows:

$$\begin{aligned}&\Delta_V L(t)\\ &\leq\mathbb{E}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[Z_i^{(c)}(t)\Big(1-\delta_i^{(c)}\Big)\Big(Q_i^{(c)}(t)+R_i^{(c)}(t)\Big)\\ &\qquad-\Big(Z_i^{(c)}(t)+Q_i^{(c)}(t)+R_i^{(c)}(t)\Big)1_i(t)\\ &\qquad\times\Big(D_{r,i}^{(c)}(t)+D_{l,i}^{(c)}(t)\Big)|\overrightarrow{U}(t)\Big]\\ &\quad-\mathbb{E}\left[\sum_{i\in\mathcal{N}}\tilde{E}_i(t)\left[1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{(c)}(t)+p_{tx,i}^{(c)}(t)\Big)-e_i(t)\right]|\overrightarrow{U}(t)\right]\\ &\quad+V\cdot\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big]+B\end{aligned} \tag{21}$$

where $B\triangleq(1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(2+\delta_i^{(c)})(R_i^{(c),\max}+Q_i^{(c),\max})^2+(1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(\max\{(p_{l,i}^{(c),\max}+p_{tx,i}^{(c),\max}),\varepsilon_i^{\max}\})^2$.

Proof: See Appendix B. ■

Inferring from Lemma 1, we can minimize the system energy consumption by minimizing the upper bound of $\Delta_V L(t)$ and simultaneously ensure the constraint of queue stability. Thus, we transform the optimization problem P2 to the P3 based on Lemma 1 as follows:

$$\begin{aligned}\text{P3}: \ \min_{\{\vec{\mathbf{u}}(t)\}} \ &VP(t)\\ &-\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[\Big(Z_i^{(c)}(t)+Q_i^{(c)}(t)+R_i^{(c)}(t)\Big)\cdot 1_i(t)\Big(D_{r,i}^{(c)}(t)\\ &\qquad+D_{l,i}^{(c)}(t)\Big)\Big]\\ &-\sum_{i\in\mathcal{N}}\tilde{E}_i(t)\left[1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{(c)}(t)+p_{tx,i}^{(c)}(t)\Big)-e_i(t)\right]\\ \text{s.t.} \ &(1), (4), (6), (12)\end{aligned}$$

where $P(t)=\sum_{i=1}^{N}\sum_{c\in\mathcal{C}}1_i(t)(p_{l,i}^{(c)}(t)+p_{tx,i}^{(c)}(t))$.

In P3, V is a constant parameter to control the tradeoff process between the system objective and constraints.

Algorithm 1: LYP-CEMA

```
foreach Time slot t do
    foreach MU i do
        if Ẽ_i(t) ≥ −V then
            MU i stays awake, 1_i(t) = 1
        else
            MU i enters sleep mode, 1_i(t) = 0.
        end
    end
    All MUs offload their states to the interconnected
    MEC system.
    The MEC system collects all commodity conditions
    and obtains the optimal solutions by solving P3.
    Commodity c downloads the result set from the MEC
    system.
end
```

As V increases, the optimal value of P3 converges to the optimal value of the original problem P1. We prove the asymptotic optimality and the optimal gap between the P3 and P1 in Theorem 4. Then, given the system input vector $\vec{\mathbf{v}}(t) = [Z_i^{(c)}(t), R_i^{(c)}(t), \tilde{E}_i(t), \varepsilon_i(t)]$, we conclude the solution for P3 in Algorithm 1.

B. Properties of Optimization Algorithm

Under LYP-CEMA, we solve the problem P3 for each time slot t, and we have the following properties of the optimal solution in Lemma 2.

Lemma 2: The optimal solution to P3 under LYP-CEMA has the properties as follows.

1) The optimal solution to the energy management: if $\tilde{E}_i(t) < -V$, then $e_i(t)^* = \varepsilon_i(t)$; if $\tilde{E}_i(t) \geq -V$, then $e_i(t)^* = 0$.
Proof: See Appendix C. ■
2) The optimal solution to the QoS constraint: if $Z_i^{(c)}(t) < V - (Q_i^{(c)}(t) + R_i^{(c)}(t))$, then $(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)) = 0$; if $Z_i^{(c)}(t) \geq V - (Q_i^{(c)}(t) + R_i^{(c)}(t))$, then $(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)) \geq (1 - \delta_i^{(c)})(Q_i^{(c)}(t) + R_i^{(c)}(t))$.

C. Performance Analysis

In this section, we first prove the feasibility of LYP-CEMA in Theorems 2 and 3. Then, we show the gap between the optimal value under LYP-CEMA and the original problem P1 in Theorem 4.

First, we define the perturbation parameter θ_i as follows:

$$\theta_i \geq \tilde{E}_i^{\max} + V \cdot \left(E_i^{\min}\right)^{-1} \tag{22}$$

where $\tilde{E}_i^{\max} = \min\{\max\{\kappa L_i (f_i^{(c),\max})^2, \tau p_{tx}^{(c),\max}\}, E_i^{\max}\}$, and we show that the required capacity of battery is determined by the control parameter V. Later, we will give the simulation results on the relationship of the required capacity of battery versus V in Section VI.

Theorem 2: The bound of battery level, $0 \leq E_i(t) \leq \theta_i + \varepsilon_i^{\max}$, is always satisfied for all i and t.

Proof: See Appendix D. ■

Theorem 3: The QoS virtual queue is always bounded by $Z_i^{(c)}(t) \leq Z_i^{(c),\max} = V + R_i^{(c),\max} + Q_i^{(c),\max}$, for all i, c, and t, which denotes the worst case of the data length in QoS virtual queue. Besides, the amount of the execution data that uncompleted timely in a period T is upper bounded by $Z_i^{(c),\max} + T\delta_i^{(c)}(R_i^{(c),\max} + Q_i^{(c),\max})$.

Proof: See Appendix E. ■

Theorem 4: We denote the achieved system average power consumption under LYP-CEMA as $\overline{P}^*$ and denote the optimal value of the original problem P1 as $\overline{P}^{\text{opt}}$. Then, under LYP-CEMA, we have

$$\overline{P}^* \leq \overline{P}^{\text{opt}} + \frac{B}{V} \tag{23}$$

where $B \triangleq (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(2+\delta_i^{(c)})(R_i^{(c),\max}+Q_i^{(c),\max})^2 + (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(\max\{(p_{l,i}^{(c),\max} + p_{tx,i}^{(c),\max}), \varepsilon_i^{\max}\})^2$.

Proof: See Appendix F. ■

Remark 1: According to Theorem 4, we have that the optimality gap of the power consumption will be smaller by choosing a larger V, i.e., the proposed algorithm can asymptotically achieve the optimal performance of the original design problem P1. Thus, by adjusting the control parameter V, we can balance the long-term system performance, the asymptotic optimality gap, and the required battery capacity.

V. ADMM-Based Distributed Algorithm Design

In this section, based on LYP-CEMA, we propose a distributed algorithm to solve the growing computational complexity problem with large-scale access requests. Besides, under the distributed algorithm, each MU makes its own control decisions to optimize the power consumption problem, instead of being entirely controlled by the system and forced to provide private information to MEC server [21]. The distributed algorithm is based on the ADMM method which can decompose the convex optimization problem into different subproblems, and each subproblem is easier to be handle iteratively and parallelly with the defined variables [16].

A. Algorithm Design

Considering the real-world strategy for maintaining a long battery life, the properties in Lemma 2 of controlling $e_i(t)^*$ as zero is not reasonable in this design. Thus, we assume that the charging level $e_i(t)^* = \varepsilon_i(t)$ continuously. Then, $\vec{\mathbf{u}}(t)$ can be reformed as $\vec{\mathbf{u}}'(t) = [\vec{\mathbf{f}}(t), \vec{\mathbf{D}}_r(t), \vec{\mathbf{p}}_{tx}(t), \vec{\boldsymbol{\alpha}}(t)]$. Making the term $VP(t)$ split and merged to the whole formula, we have the reformed version of P3, P3′ as follows:

$$\begin{aligned}\text{P3}' :\ \min_{\{\mathbf{u}'(t)\}}\ & V\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}} 1_i(t)\left[\kappa\left[f_i^{(c)}(t)\right]^2 + p_{tx,i}^{(c)}(t)\right] \\ & - \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\left[\left(Z_i^{(c)}(t) + Q_i^{(c)}(t) + R_i^{(c)}(t)\right)\right. \\ & \left.\times\ 1_i(t)\left(\tau f_i^{(c)}(t)L_i^{-1} + D_{r,i}^{(c)}(t)\right)\right] \\ & - \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\tilde{E}_i(t)\left[1_i(t)\left(\kappa\left[f_i^{(c)}(t)\right]^2 + p_{tx,i}^{(c)}(t)\right)\right] \\ \text{s.t.}\ & (1), (4), (6), (12).\end{aligned}$$

Then, to implement the distributed algorithm framework, we define a set of the four control variables as $\vec{x}_i^{(c)}$, which means $\vec{x}_i^{(c)}(1) = f_i^{(c)}(t), \vec{x}_i^{(c)}(2) = D_{r,i}^{(c)}(t), \vec{x}_i^{(c)}(3) = p_{tx,i}^{(c)}(t)$, and $\vec{x}_i^{(c)}(4) = \alpha_i^{(c)}(t)$. Meanwhile, a new optimization vector $\mathbf{x} \triangleq [\vec{x}_1^{(1)}, \ldots, \vec{x}_1^{(C)}, \ldots, \vec{x}_N^{(1)}, \ldots, \vec{x}_N^{(C)}]$ is defined and overall 4NC variables in it. Therefore, we can sort out the P3′ as the following formula regarding to the variable $\vec{x}_i^{(c)}$:

$$\begin{aligned} \text{P4}: \ & \min_{\mathbf{x}} \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\left[F_i\left(\vec{x}_i^{(c)}\right)\right] \\ & \text{s.t.} \ \vec{x}_i^{(c)} \in \mathcal{X}, \forall i, c \end{aligned}$$

where

$$\begin{aligned} F_i\left(\vec{x}_i^{(c)}\right) = & \left[\left(V - \tilde{E}_i(t)\right)1_i(t)\kappa\right]\left(\vec{x}_i^{(c)}(1)\right)^2 \\ & - \left[\left(Z_i^{(c)}(t) + Q_i^{(c)}(t) + R_i^{(c)}(t)\right)1_i(t)\tau L_i^{-1}\right]\vec{x}_i^{(c)}(1) \\ & - \left[\left(Z_i^{(c)}(t) + Q_i^{(c)}(t) + R_i^{(c)}(t)\right)1_i(t)\right]\vec{x}_i^{(c)}(2) \\ & + \left[\left(V - \tilde{E}_i(t)\right)1_i(t)\right]\vec{x}_i^{(c)}(3) \end{aligned} \tag{24}$$

and the constraints set $\mathcal{X}$ is derived from constraints (1), (4), and (6), i.e., $\mathcal{X} \triangleq \{[0, f_i^{(c),\max}], [0, D_{r,i}^{(c),\max}], [0, p_{tx}^{(c),\max}], \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}} \alpha_i^{(c)}(t) = 1, \}$, where $D_{r,i}^{(c),\max} = w\tau\log_2(1 + [(Hp_{tx,i}^{(c),\max})/N])$.

According to (5), we have the equation $\vec{x}_i^{(c)}(2) = \vec{x}_i^{(c)}(4)w\tau\log_2(1+[(H\vec{x}_i^{(c)}(3))/N])$ to represent the relationship between the three variables. With the problem of bandwidth allocation in constraint (6), we decide the bandwidth allocation of MU i depending on the amount of data that transmitted to MEC server and the channel status. Later, we will show the simulation result of bandwidth allocation in Section VI to make the further illustration.

Next, we introduce an assistant vector $\mathbf{z}$ as a copy of $\mathbf{x}$. And P4 can be reformulated into an equivalent problem in the format of ADMM method, P5, as follows:

$$\begin{aligned} \text{P5}: \ & \min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ & \text{s.t.} \ \mathbf{x} - \mathbf{z} = 0 \end{aligned} \tag{25}$$

where

$$\begin{aligned} f(\mathbf{x}) = \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Bigg[F_i\left(\vec{x}_i^{(c)}\right) + \mathbf{1}\Bigg(\vec{x}_i^{(c)} \in \mathcal{X}, \vec{x}_i^{(c)}(2) = \vec{x}_i^{(c)}(4)w\tau \\ \times \log_2\left(1 + \frac{H\vec{x}_i^{(c)}(3)}{N}\right)\Bigg)\Bigg] \end{aligned}$$

$$g(\mathbf{z}) = \mathbf{1}\left(\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\vec{z}_i^{(c)}(4) = 1\right)$$

and $\mathbf{1}(\cdot)$ is an indicator function that if the object in brackets is true, it equals 0 and infinity otherwise. Through the above transformation, the reformulated problem P5 can adopt ADMM optimization method and the distributed optimization algorithm can be developed.

Next, we define a dual variable $\mathbf{y} \triangleq [\vec{y}_1^{(1)}, \ldots, \vec{y}_1^{(C)}, \ldots, \vec{y}_N^{(1)}, \ldots, \vec{y}_N^{(C)}]$ with the ADMM approach [16] to complete the following iteration calculation. We denote $x_i^{(c),k}$, $z_i^{(c),k}$, and $y_i^{(c),k}$ as the respective variables at the kth iteration. We update the iteration variables values as follows:

$$\begin{aligned} x_i^{(c),k+1} = \arg\min_{x_i}\Bigg\{ & F_i\left(\vec{x}_i^{(c)}\right) + \frac{\rho}{2}\left\|\vec{x}_i^{(c)} - z_{i,j}^{(c),k} + \frac{y_i^{(c),k}}{\rho}\right\|_2^2 \\ & \Bigg| \vec{x}_i^{(c)} \in \mathcal{X}, \vec{x}_i^{(c)}(2) = \vec{x}_i^{(c)}(4)w\tau\log_2 \\ & \times \left(1 + \frac{H\vec{x}_i^{(c)}(3)}{N}\right)\Bigg\}, \forall i, c \end{aligned} \tag{26}$$

$$\begin{aligned} \mathbf{z}^{k+1} = \arg\min_{\mathbf{z}}\Bigg\{ & \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\left\|\vec{z}_i^{(c)} - \frac{y_i^{(c),k}}{\rho} - x_i^{(c),k+1}\right\|_2^2 \\ & \Bigg|\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\vec{z}_i^{(c)}(4) = 1\Bigg\} \end{aligned} \tag{27}$$

$$y_i^{(c),k+1} = y_i^{(c),k} + \rho\left(x_i^{(c),k+1} - z_i^{(c),k+1}\right), \forall i, c \tag{28}$$

where the penalty parameter $\rho > 0$ is a step size and it needs to be chosen appropriately, then $x_i^{(c),k+1}$ and $\mathbf{z}^{k+1}$ will converge to an optimal point and $y_i^{(c),k+1}$ will converge to an optimal dual point.

B. Optimality Conditions and Stopping Criterion

In this section, we show that the optimality conditions and stopping criterion of the distributed algorithm and make an analysis on the convergence rate of our proposed algorithm.

First, we denote the optimal primal variables as $\mathbf{x}^*$ and $\mathbf{z}^*$, and the optimal dual variable as $\mathbf{y}^*$. Meanwhile, the optimal value of P5 can be expressed as $f(\mathbf{x}^*) + g(\mathbf{z}^*)$. According to [16], we can get the primal feasibility condition

$$\mathbf{x}^* - \mathbf{z}^* = 0 \tag{29}$$

and the dual feasibility conditions

$$0 \in \partial f\left(\mathbf{x}^*\right) + \mathbf{y}^* \tag{30}$$

$$0 \in \partial g\left(\mathbf{z}^*\right) - \mathbf{y}^*. \tag{31}$$

Based on the above optimality conditions, we can derive the convergence measures for the distributed algorithm based on ADMM method iterates $(x_i^{(c),k}, z_i^{(c),k}, y_i^{(c),k})$.

Next, we introduce the stopping criterion of the algorithm. According to the analysis in the optimality conditions, we denote $r^{k+1} = x_i^{(c),k+1} - z_i^{(c),k+1}$ as the primal residual which is a convergence measure of primal feasibility based on (30) and $s^{k+1} = -\rho(z_i^{(c),k+1} - z_i^{(c),k})$ as the dual residual based on the dual feasibility condition at iteration $k+1$. And we need to prescribe a reasonable termination criterion, which is the stop condition of the iterations when the primal and dual residuals satisfy

$$\left\|r^k\right\|_2 \le \epsilon^{\text{pri}} \quad \text{and} \quad \left\|s^k\right\|_2 \le \epsilon^{\text{dual}} \tag{32}$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual conditions, respectively.

Algorithm 2: ADMM-DEMA

foreach Time slot t **do**
  foreach MU i **do**
    if $\tilde{E}_i(t) \geq -V$ **then**
      MU i stays awake, $1_i(t) = 1$.
    else
      MU i enters sleep mode, $1_i(t) = 0$.
    end
    All MUs send their signals to the interconnected MEC system.
  end
  repeat
    foreach Commodity c of awake MU i **do**
      Commodity c individually solves (26) and offloads its system states and solutions $x_i^{(c),k+1}$ to MEC.
    end
    The MEC system collects all commodity conditions and obtains $\mathbf{z}^{k+1}$ by solving (27).
    foreach Commodity c of awake MU i **do**
      Commodity c downloads the result set and updates $y_i^{(c),k+1}$ to MU i and MEC.
    end
  until *(32) is TRUE*;
end

Remark 2: According to the theorems in [22], we can show that our proposed algorithm based on ADMM method has a convergence rate $O(1/k)$. Compared with the convergence rate of subgradient-based algorithm is $O(1/\sqrt{k})$, our proposed algorithm is faster and can be used well in real time implementation. Note that the proof is to satisfy the two optimality assumptions of ADMM in [16]. Details are omitted.

C. Implementation

In this section, we show the implementation of the distributed algorithm based on ADMM method.

First, we consider that the MEC server also works as a central aggregator for the execution and communication of MUs. At the iteration $k = 0$, we initialize the data. Next, at $k = 1$, each commodity c of MU i can individually solve the minimization problem (26) and sends $z_i^{(c),k}$ and the updated value, $x_i^{(c),k+1}$ to the MEC server. Next, the MEC server updates $\mathbf{z}^{k+1}$ by solving (27) and broadcasting the results back to each MU. Then, each commodity c of MU i updates $y_i^{(c),k+1}$ according to (28) and the primal and dual residuals. Until the stop conditions (32) are satisfied, the iterations stop and we get the optimum under the distributed algorithm.

In the actually implementation of the distributed algorithm, the two residuals eventually will converge to 0 and the dual variable $y_i^{(c),k}$ will converge to the dual optimal point as iteration $k \to \infty$. The residual convergence shows the iterates approach feasibility. Meanwhile, the problem P5 will converge to the optimum under the ADMM optimization method when the stop conditions (32) are satisfied with iteration $k \to \infty$ [16]. We conclude ADMM-DEMA in Algorithm 2.

VI. Empirical Evaluation

In this section, we evaluate the performance of the proposed algorithms through extensive simulations. We verify the feasibility and the efficiency of our proposed algorithms on optimizing the power consumption minimization problem.

A. Setup

In simulations, we consider 4 MUs interconnected to the MEC system operated on discrete time slots which $\tau = 1$ s. Each MU has installed five commodities and a dual-core CPU with maximum frequency $f_i^{(c),\max} = 2.4$ GHz. The offloading channel power gain $H = -117$ dB. The noise power $N_0 = -100$ dBm and bandwidth $w = 20$ MHz. For local execution, the maximum new data processing requests $R_i^{(c),\max} = 15$ Mb and the maximum quality data residue $Q_i^{(c),\max} = 35$ Mb. Each battery has a maximum capacity $E_i^{\max} = 65$ Wh and the harvested energy arrivals are bounded by $\varepsilon_i^{\max} = 1.5$ Wh. Also, $R_i^{(c)}(t)$ and $e_i(t)$ are uniformly distributed within $[0, R_i^{(c),\max}]$ and $[0, \varepsilon_i(t)]$, respectively. In addition, several other parameters are set as $\kappa = 7.5 \times 10^{-4}$, $L_i = 64$, $\delta_i^{(c)} = 0.2$, and $p_{tx}^{(c),\max} = 0.2$ Wh.

B. Performance Evaluation

In Fig. 3, we show the relationship between the time-average system power consumption of MUs in different algorithms and the required battery capacity, $E_i^{\max}$, versus the parameter V. For LYP-CEMA and ADMM-DEMA, the time-average system power consumption $\overline{P}$ declines rapidly and then converges to the suboptimal value when $V > 500V_{\max}$ as shown in Fig. 3(a). Besides, we employ the Greedy algorithm to be the benchmark algorithm for comparison. In Greedy algorithm, the system always optimizes the power consumption at the current time slot and considers no future battery and queue dynamic information. According to this property, the Greedy algorithm usually can obtain the approximate optimal solution for most optimization problems, but the result is only local optimal, because the greedy algorithm does not consider the overall optimality of the problem. As shown in Fig. 3(a), we can reduce much power consumption when using LYP-CEMA and ADMM-DEMA compared with the Greedy algorithm. Moreover, by comparing the system performance between the distributed algorithm and the centralized algorithm, we find that the time-average system power consumption under LYP-CEMA declines faster than ADMM-DEMA in the beginning. And with V increasing, both converge to the suboptimal value eventually. This reveals that the efficiency of ADMM-DEMA is close to LYP-CEMA.

The required battery capacity in ADMM-DEMA increases linearly with V as shown in Fig. 3(b). Note that in Fig. 3(b), the battery capacity is impractical when V is upper than $500V_{\max}$ and we just show it to imply the relationship between battery capacity and parameter V. According to Fig. 3, it shows the efficiency of the proposed algorithms and implicates that it achieves the suboptimality of system performance under the algorithms which proved in Theorem 4.

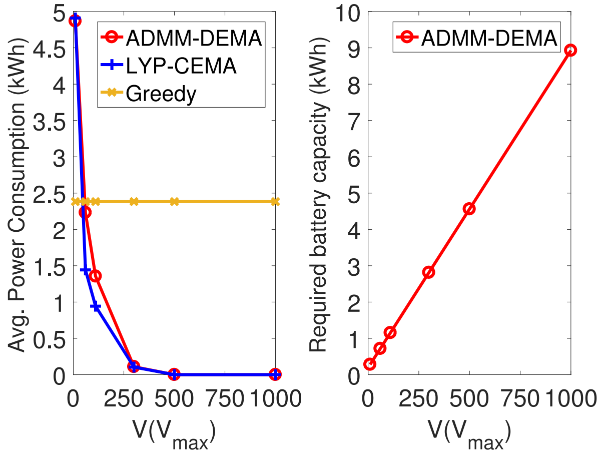


Fig. 3. Average power consumption of MUs $\overline{P}$/required battery capacity versus V.

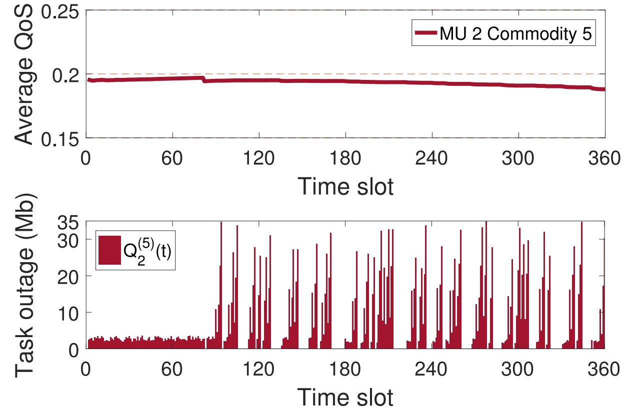


Fig. 5. Average QoS with $\delta_2^{(5)} = 0.2$ and the quality execution data residue $Q_2^{(5)}(t)$ for MU 2 commodity 5 versus time slot.

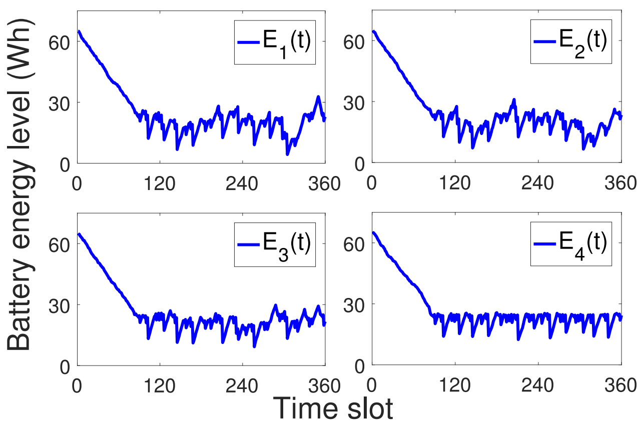


Fig. 4. Battery energy level $E_i(t)$ for MUs versus time slot.

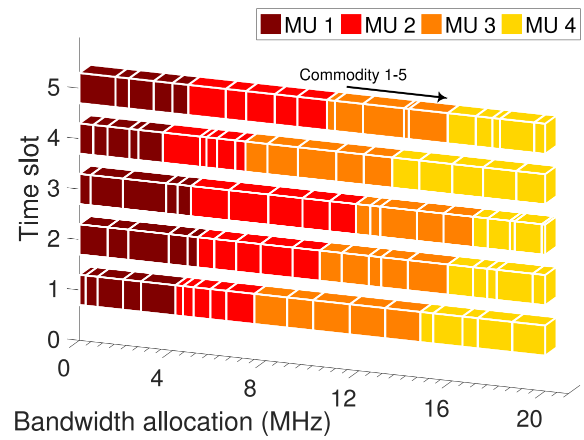


Fig. 6. Bandwidth allocation $\alpha_i^{(c)}(t)$ for 5 commodities of 4 MUs versus time slot.

C. Feasibility Verification

In this section, we verify the feasibility of the proposed algorithms on the stability of battery queue and QoS queue which is discussed in Theorems 2 and 3. We also analyze bandwidth allocation and the situation of several main variables during the system running.

We show the battery energy level $E_i(t)$ for MUs in Fig. 4. It demonstrates that in different MU i, the battery level decreases over time at the start since the continuous charging of renewable energy and stabilizes gradually around the perturbed energy level. This verifies the stability of the battery queue and the conclusion in Theorem 2: the battery level constraint, $0 \le E_i(t) \le E_i^{\max}$, is always satisfied for all i and t.

Next, we show the simulation results of the average QoS with the probability $\delta_2^{(5)} = 0.2$ and take the quality execution data backlog $Q_2^{(5)}(t)$ of MU 2 commodity 5 as an instance in Fig. 5(a). It shows that the average QoS is always lower than the prescribed probability $\delta_2^{(5)} = 0.2$ under ADMM-DEMA, which verifies the conclusion in Theorem 1. Besides, the change over time of the data queue, $Q_2^{(5)}(t)$, has been shown in Fig. 5(b). It can be seen that the backlog is stable at beginning and increases gradually with the proceeding of the system. Note that at the time slots when $Q_2^{(5)}(t)$ exceeds the upper bound $Q_2^{(5),\max} = 35$ Mb, MU dropped the all requests in $Q_i^{(c)}(t)$ as shown in Fig. 5(b) to ensure the QoS constraint.

Then, we present the bandwidth allocation for 5 commodities of 4 MUs at first 5 time slots in Fig. 6. In each time slot, 20 commodities share the total bandwidth 20MHz and it also shows the feasibility of our proposed algorithms. In Fig. 6, we can see that in each time slot, the bandwidth allocation is varied based on the amount of data request and the channel status of different commodity c of MU i.

Finally, in Fig. 7, we plot the simulation results of the main variables for MU 2 commodity 5 under ADMM-DEMA. The left column of Fig. 7 is related to MEC server execution model and the transmission power consumption $p_{tx,2}^{(5)}(t)$ is always equal to 0.2Wh for all time slots t under ADMM-DEMA. The trend of the bandwidth allocation $\alpha_2^{(5)}(t)w$ and the offloaded execution data $D_{r,2}^{(5)}(t)$ is similar and this reveals that the relationship between $D_{r,2}^{(5)}(t)$ and $\alpha_2^{(5)}(t)$ in (5). Then, the right column of Fig. 7 is related to local execution model, it also reveals that the relationship between the local executed data size $D_{l,2}^{(5)}(t)$ and local execution power consumption $p_{l,2}^{(5)}(t)$ with the CPU frequency $f_2^{(5)}(t)$ in (2) and (3) by comparison between figures. In general, Fig. 7 verifies the feasibility of

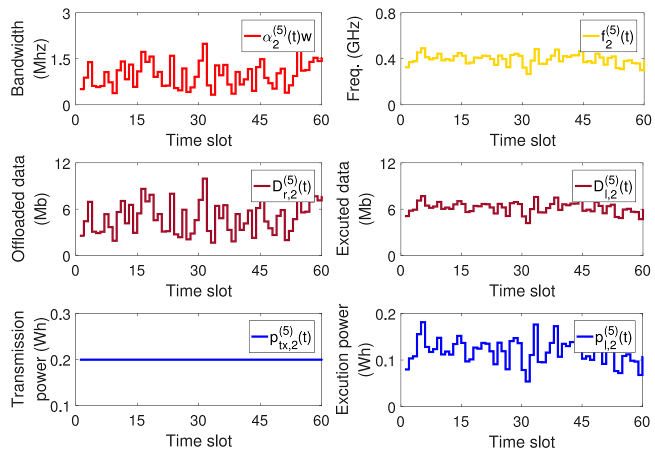


Fig. 7. Bandwidth allocation $\alpha_2^{(5)}(t)$, CPU frequency $f_2^{(5)}(t)$, offloaded/local executed data size $D_{r,2}^{(5)}(t)$, $D_{l,2}^{(5)}(t)$, and transmission/local execution power consumption $p_{tx,2}^{(5)}(t)$, $p_{l,2}^{(5)}(t)$ versus time slot.

ADMM-DEMA and shows the actual operation of the MEC system under ADMM-DEMA.

VII. Related Work

The contradiction between the limited capabilities of mobile devices and the growing demand of users is still a major obstacle to the development of IoT applications. To build scalable IoT devices and platforms is becoming the research focus [23], [24]. MEC technology has evolved as a promising technique to relieve the computation stress and save energy by offloading the execution tasks of MUs to the MEC server [25].

Energy management in MEC systems has been extensively studied in [26]–[28]. A dynamic offloading algorithm based on Lyapunov optimization was developed to tradeoff the energy and the execution time in [26]. For minimizing the energy consumption, the optimization problem under the constraint of the edge cloud computation capacities and the execution latency was formulated in [28]. Besides, in [29], to maximize the network utility by balancing throughput and fairness, A perturbed Lyapunov function is designed and a knapsack problem is solved per slot for the optimal schedule. The above mentioned works show the importance of the research on energy management in MEC systems. In this paper, we investigate the energy management problem for MEC systems, and meanwhile, we consider the MUs equipped with EH devices and the QoS constraint.

A few recent works have started to consider the multiuser scenarios [30], [31]. In [30], it jointly optimized the radio resource scheduling and computation offloading in order to minimize the average energy consumed by all the users terminals. In [31], it studied the multiuser computation offloading problem for mobile-edge cloud computing in a multichannel wireless interference environment. In summary, the study of MEC system in multiuser scenarios is more realistic and the results obtained are more valuable.

The MEC systems with renewables has been studied in [20], [32], and [33]. In [20], an effective computation offloading strategy was developed in a green MEC system with EH devices. In [32], an efficient reinforcement learning-based resources management algorithm in EH mobile edge computing was proposed. In [33], each IoT device is equipped with an EH module to harvest the ambient energy to support the sustainable operation. EH technology is becoming widely applied in the communication systems with its self sustaining supply of energy and green communication [34], [35]. In this paper, we investigate the energy management in MEC systems with EH devices. We model that each MU is equipped with EH devices and assume that the system is only powered by the harvesting energy.

Several optimization methods have been adopted in the recent works on MEC technology [36]–[38]. In [37], a joint computation allocation and resource management algorithm was proposed based on Lyapunov optimization theory. By leveraging Lyapunov optimization technique, the proposed algorithms work online, and achieve an asymptotically optimal performance. However, most of these algorithms are centralized, which means that they usually need a center to process the data from the entire network. To reduce the computational burden, some optimization methods related to the distributed computation can be utilized, including the auxiliary problem principle and the Lagrangian relaxation. In this paper, we first design an online algorithm based on Lyapunov optimization method. Furthermore, to reduce the system computational complexity, we propose a distributed algorithm based on ADMM method [16]. By using this method, the convex optimization problem can be decomposed into different subproblems, and each subproblem is easier to handle iteratively and parallelly with the defined variables [16]. Meanwhile, the method has the advantages of simplicity, fast convergence, and low computational complexity and is well suited to distributed convex optimization, in particular to large-scale problems [16].

Compared to the existing studies, in this paper, we consider a multiuser MEC system with EH devices to investigate the energy management problem of MEC systems. The formulated power minimization problem subjects to the operational constraint, QoS constraint and the heavy computation due to the centralized control of multiuser. To overcome the challenges, we develop the Lyapunov optimization-based online algorithm and the distributed algorithm based on ADMM method.

VIII. Conclusion

In this paper, we investigated the power consumption minimization problem for a multiuser MEC system with EH devices. A power consumption minimization problem with the battery queue stability constraint and the QoS constraint was formulated as an asymptotic optimization problem. We propose an online algorithm based on Lyapunov optimization method to solve the problem. The implementation of the online algorithm only depends on the information of the current moment. Furthermore, we developed a distributed algorithm based on the ADMM method to reduce the computational complexity. Performance analysis was conducted which shows the asymptotic optimality of the proposed algorithms and the upper bound of the QoS queue and battery queue. Simulation results verify the theoretical analysis and the effectiveness of the proposed algorithms for optimizing the power consumption.

Appendix A
Proof of Theorem 1

According to the QoS dynamics, we have

$$\begin{cases} Z_i^{(c)}(1) \geq Z_i^{(c)}(0) - \delta_i^{(c)}\Big[Q_i^{(c)}(0) + R_i^{(c)}(0)\Big] + Q_i^{(c)}(1) \\ \ldots \\ Z_i^{(c)}(t) \geq Z_i^{(c)}(t-1) \\ \qquad -\delta_i^{(c)}\Big[Q_i^{(c)}(t-1) + R_i^{(c)}(t-1)\Big] + Q_i^{(c)}(t). \end{cases}$$

Summing up the above inequalities, we have

$$Z_i^{(c)}(t) \geq Z_i^{(c)}(0) - \delta_i^{(c)}\sum_{\tau=0}^{t-1}\Big[Q_i^{(c)}(\tau) + R_i^{(c)}(\tau)\Big] + \sum_{\tau=1}^{t} Q_i^{(c)}(\tau).$$

Dividing both sides by t and letting t go to infinity, we have

$$\begin{aligned} &\lim_{t\to\infty}\frac{Z_i^{(c)}(t) - Z_i^{(c)}(0)}{t} \\ &\geq \lim_{t\to\infty}\frac{1}{t}\Bigg[\Big(1-\delta_i^{(c)}\Big)\sum_{\tau=0}^{t-1}\Big[Q_i^{(c)}(\tau) + R_i^{(c)}(\tau)\Big] \\ &\quad - \lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1} 1_i(t)\Big(D_{r,i}^{(c)}(\tau) + D_{l,i}^{(c)}(\tau)\Big). \end{aligned}$$

Note that $Z_i^{(c)}(0)$ is finite. If we make the QoS virtual queue $Z_i^{(c)}(t)$ stable by a control policy, it is finite for all t. Then we have $\lim_{t\to\infty}[(Z_i^{(c)}(t) - Z_i^{(c)}(0))/t] = 0$, which deduces $\rho_i^{(c)} \leq \delta_i^{(c)} \cdot \lambda_i^{(c)}$ with the definition of $\lambda_i^{(c)}$ and $\rho_i^{(c)}$.

Appendix B
Proof of Lemma 1

First, squaring both sides of the QoS virtual queue dynamics, summing over $i \in \mathcal{N}$, and multiplying both sides by $(1/2)$, we have

$$\begin{aligned} &\frac{1}{2}\mathbb{E}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[Z_i^{(c)}(t+1)^2 - Z_i^{(c)}(t)^2\Big] \\ &\leq B_1 + \mathbb{E}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[Z_i^{(c)}(t)\Big(1-\delta_i^{(c)}\Big)\Big(Q_i^{(c)}(t) + R_i^{(c)}(t)\Big) \\ &\quad - \Big(Z_i^{(c)}(t) + Q_i^{(c)}(t) + R_i^{(c)}(t)\Big)1_i(t)\Big(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)\Big)\Big] \end{aligned} \tag{33}$$

where $B_1 \triangleq (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(2+\delta_i^{(c)})(R_i^{(c),\max} + Q_i^{(c),\max})^2$.

Similarly, squaring both sides of the battery virtual queue dynamics, summing over $i \in \mathcal{N}$, and multiplying both sides by $(1/2)$, we have

$$\begin{aligned} &\frac{1}{2}\mathbb{E}\sum_{i\in\mathcal{N}}\Big[\tilde{E}_i(t+1)^2 - \tilde{E}_i(t)^2\Big] \\ &\leq B_2 - \mathbb{E}\sum_{i\in\mathcal{N}}\Bigg[\tilde{E}_i(t)\cdot 1_i(t)\sum_{c\in\mathcal{C}}\Big[p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big] - e_i(t)\Bigg] \end{aligned} \tag{34}$$

where $B_2 \triangleq (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(\max\{(p_{l,i}^{(c),\max} + p_{tx,i}^{(c),\max}), \varepsilon_i^{\max}\})^2$.

We define the constants in the right part of the two inequalities, $B_1 + B_2$, as

$$\begin{aligned} B &\triangleq \frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big(2+\delta_i^{(c)}\Big)\Big(R_i^{(c),\max} + Q_i^{(c),\max}\Big)^2 \\ &\quad + \frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big(\max\Big\{\Big(p_{l,i}^{(c),\max} + p_{tx,i}^{(c),\max}\Big), \varepsilon_i^{\max}\Big\}\Big)^2. \end{aligned}$$

Then, by summing (33) and (34), taking expectation on both sides conditioning on $U(t)$. And using the definition of $\Delta_v L(t)$, Adding the term $V \cdot \mathbb{E}[P(t)|\overrightarrow{U}(t)]$ to it, we have

$$\begin{aligned} &\Delta_V L(t) \\ &\leq \mathbb{E}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[Z_i^{(c)}(t)\Big(1-\delta_i^{(c)}\Big)\Big(Q_i^{(c)}(t) + R_i^{(c)}(t)\Big) \\ &\qquad - \Big(Z_i^{(c)}(t) + Q_i^{(c)}(t) + R_i^{(c)}(t)\Big)1_i(t) \\ &\qquad \times \Big(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)\Big)|\overrightarrow{U}(t)\Big] \\ &\quad - \mathbb{E}\Bigg[\sum_{i\in\mathcal{N}}\tilde{E}_i(t)\Bigg[1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big) - e_i(t)\Bigg]\Bigg|\overrightarrow{U}(t)\Bigg] \\ &\quad + V\cdot\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big] + B \end{aligned}$$

where $B \triangleq (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(2+\delta_i^{(c)})(R_i^{(c),\max} + Q_i^{(c),\max})^2 + (1/2)\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}(\max\{(p_{l,i}^{(c),\max} + p_{tx,i}^{(c),\max}), \varepsilon_i^{\max}\})^2$.

We see that the above completes the proof of Lemma 1.

Appendix C
Proof of Lemma 2

In the previous paper, we develop the initial problem to P3 by Lyapunov Optimization method. We can get the optimal solution by solving P3 includes the optimal EH, i.e., to determine $e_i(t)$ and the optimal execution decision, i.e., to determine $\vec{\mathbf{f}}(t)$, $\vec{\mathbf{p}}_{tx}(t)$, and $\vec{\boldsymbol{\alpha}}(t)$.

Using the partitioning method to solve P3, we can obtain the optimal amount of energy that charged to battery of MU i at time t, $e_i(t)^*$, by solving the following problem:

$$\begin{aligned} \min\ & \tilde{E}_i(t)e_i(t) \\ \text{s.t.}\ & e_i(t) \in [0, \varepsilon_i(t)] \end{aligned} \tag{35}$$

and we have its optimal solution is given by

$$e_i(t)^* = \varepsilon_i(t)\cdot\mathbf{1}\big(\tilde{E}_i(t) < 0\big). \tag{36}$$

According to (36), we can easily obtain Lemma 2.

Appendix D
Proof of Theorem 2

According to the algorithm and θ_i, we have, $E_i^{\max} \triangleq \theta_i + \varepsilon^{\max}$, i.e., $0 \leq E_i(t) \leq \theta_i + \varepsilon^{\max}$. To prove Theorem 1, we first show the following properties for the optimal scheduling.

Using the Lemma 2.(1), we can show the proof of the constraint (9) as follows. First, we can get the lower bound of $E_i(t)$ directly as that the $E_i(t) \geq 1_i(t)\sum_{c\in\mathcal{C}}(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t))$ always holds under the algorithm. Then, the upper bound of $E_i(t)$ can be shown based on the Lemma 1.

We assume that $E_i(t) \leq \theta_i + \varepsilon^{\max}$, and the next step we need to show is that $E_i(t+1) \leq \theta_i + \varepsilon^{\max}$. We discuss the two cases of $E_i(t)$ as follows.

1) $E_i(t) < \theta_i$, i.e., $\tilde{E}_i(t) < 0$. Using Lemma 2.(1) and the iteration of $E_i(t)$, since $e_i(t)^* = \varepsilon_i(t)$, we have $E_i(t+1) \leq E_i(t) + e_i(t)^* \leq \theta_i + e_i(t)^* \leq \theta_i + \varepsilon^{\max}$, i.e., $E_i(t+1) \leq \theta_i + \varepsilon^{\max}$.
2) $E_i(t) \geq \theta_i$, i.e., $\tilde{E}_i(t) \geq 0$. Using Lemma 2.(1) and the iteration of $E_i(t)$, since $e_i(t)^* = 0$, we have $E_i(t+1) \leq E_i(t) \leq \theta_i + \varepsilon^{\max}$, i.e., $E_i(t+1) \leq \theta_i + \varepsilon^{\max}$.

In summary, we have that the battery level constraint, $0 \leq E_i(t) \leq E_i^{\max}$, is always satisfied for all i and t.

Appendix E
Proof of Theorem 3

1) We first prove the $Z_i^{(c)}(t) \leq Z_i^{(c),\max}$. Above all, we have $Z_i^{(c)}(0) = 0 \leq V + R_i^{(c),\max} + Q_i^{(c),\max}$. Next, we assume that the backlog of the Qos virtual queue $Z_i^{(c)}(t) \leq Z_i^{(c),\max} = V + R_i^{(c),\max} + Q_i^{(c),\max}$ in time slot t for all i, c. Then, we prove that $Z_i^{(c)}(t+1) \leq Z_i^{(c),\max}$ still holds true in time slot $t+1$.

a) If $Z_i^{(c)}(t) > V$, then according to the Lemma 2.(2), we have $(D_{r,i}^{(c)}(t) + D_{l,i}^{(c)}(t)) \geq (1-\delta_i^{(c)})(Q_i^{(c)}(t) + R_i^{(c)}(t))$. With the Qos virtual queue dynamics, we have

$$Z_i^{(c)}(t+1) \leq \Big[Z_i^{(c)}(t) - \delta_i^{(c)}\Big(Q_i^{(c)}(t) + R_i^{(c)}(t)\Big)\Big]^+ + \delta_i^{(c)}\Big(Q_i^{(c)}(t) + R_i^{(c)}(t)\Big). \tag{37}$$

Under this condition, if $Z_i^{(c)}(t) \geq \delta_i^{(c)}(Q_i^{(c)}(t) + R_i^{(c)}(t))$, with the dynamics, we have $Z_i^{(c)}(t+1) \leq Z_i^{(c)}(t) \leq Z_i^{(c),\max}$; Otherwise, we have $Z_i^{(c)}(t+1) \leq \delta_i^{(c)}(Q_i^{(c)}(t) + R_i^{(c)}(t)) \leq Z_i^{(c),\max}$.

b) If $Z_i^{(c)}(t) \leq V$, we have $Z_i^{(c)}(t+1) \leq [Z_i^{(c)}(t) - \delta_i^{(c)}(Q_i^{(c)}(t) + R_i^{(c)}(t))]^+ + (Q_i^{(c),\max} + R_i^{(c),\max})$.

Under this condition, if $Z_i^{(c)}(t) \geq \delta_i^{(c)}(Q_i^{(c)}(t) + R_i^{(c)}(t))$, with the dynamics, we have $Z_i^{(c)}(t+1) \leq Z_i^{(c)}(t) - \delta_i^{(c)}(Q_i^{(c)}(t) + R_i^{(c)}(t)) + (Q_i^{(c),\max} + R_i^{(c),\max}) \leq V + R_i^{(c),\max} + Q_i^{(c),\max}$; Otherwise, we have $Z_i^{(c)}(t+1) \leq Q_i^{(c),\max} + R_i^{(c),\max} \leq V + R_i^{(c),\max} + Q_i^{(c),\max}$.

Overall, we show $Z_i^{(c)}(t+1) \leq Z_i^{(c),\max} = V + R_i^{(c),\max} + Q_i^{(c),\max}$. Thus, the proof of upper bound of the QoS virtual queue is finished.

2) Next, we prove the total quality execution data residue in a period T is upper bounded by $Z_i^{(c),\max} + T\delta_i^{(c)}(R_i^{(c),\max} + Q_i^{(c),\max})$. We assume a period of time $[t_1, t_2]$ with the length of T. Summing the dynamics (17) from t_1 to t_2, we have

$$\begin{aligned} &Z_i^{(c)}(t_2) \\ &\geq Z_i^{(c)}(1) - \delta_i^{(c)} \sum_{\tau=t_1}^{t_2}\Big[Q_i^{(c)}(\tau) + R_i^{(c)}(\tau)\Big] + \sum_{\tau=t_1}^{t_2} Q_i^{(c)}(\tau+1) \\ &\geq -\delta_i^{(c)} T\Big[Q_i^{(c),\max} + R_i^{(c),\max}\Big] + \sum_{\tau=t_1}^{t_2} Q_i^{(c)}(\tau+1). \end{aligned} \tag{38}$$

And the proof is completed as follows:

$$\sum_{\tau=t_1}^{t_2} Q_i^{(c)}(\tau+1) \leq Z_i^{(c)} + \delta_i^{(c)} T\Big[Q_i^{(c),\max} + R_i^{(c),\max}\Big]. \tag{39}$$

Appendix F
Proof of Theorem 4

According to Theorem 1, we have that the battery capacity constraint, $0 \leq E_i(t) \leq E_i^{\max}$ is satisfied with the control policy for all i and t. Then, taking expectation on battery dynamics and summing it over $t \in [0, T-1]$, we have

$$\begin{aligned} &\mathbb{E}\{E_i(t)\} - \mathbb{E}\{E_i(0)\} \\ &= \sum_{t=0}^{T-1}\left[\mathbb{E}\{e_i(t)\} - \mathbb{E}\left\{1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big)\right\}\right], \forall i, c. \end{aligned} \tag{40}$$

With $0 \leq E_i(t) \leq E_i^{\max}$, we divide both sides of (40) by T and taking a limit as $T \to \infty$, we have

$$\begin{aligned} &\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{e_i(t))\} \\ &= \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\{1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{(c)}(t) + p_{tx,i}^{(c)}(t)\Big)\right\}, \forall i, c. \end{aligned} \tag{41}$$

Considering the above equalities, we get a relaxed version of P1

$$\begin{aligned} \mathrm{P1}': \ &\min_{\vec{\mathbf{f}}(t), \vec{\mathbf{p}}_{tx}(t), \vec{\boldsymbol{\alpha}}(t)} \ \overline{P} \\ &\text{s.t.} \ \ (1), (4), (6), (8), (12), (38). \end{aligned}$$

Due to the constraints in P1$'$ have been relaxed, i.e., the battery energy levels can not decide the solution of P1, so the optimal solution of P1 is also feasible for P1$'$. Then we denoted the optimal solution for P1$'$ as $\{\hat{\vec{\mathbf{f}}}(t), \hat{\vec{\mathbf{p}}}_{tx}(t), \hat{\vec{\boldsymbol{\alpha}}}(t)\}$, and the object value is $\hat{\overline{P}} = \mathbb{E}\{\hat{P}(t)\} \leq \overline{P}^{\text{opt}}$. According to the properties of optimality of stationary and randomized policies, the optimal solution satisfies $\mathbb{E}[1_i(t)\sum_{c\in\mathcal{C}}(\hat{p}_{l,i}^{(c)}(t) + \hat{p}_{tx,i}^{(c)}(t)) - \hat{e}_i(t)] = 0$ and $\hat{\overline{P}} = \mathbb{E}[\hat{P}(t)]$.

We replace the optimal solution for P1′, $\{\hat{\hat{\mathbf{f}}}(t), \hat{\hat{\mathbf{p}}}_{tx}(t), \hat{\hat{\boldsymbol{\alpha}}}(t)\}$, into the right-hand-side of the *drift-plus-penalty*, we obtain

$$\begin{aligned}
&\Delta_V L(t)\\
&= \Delta L(t) + V\cdot\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big]\\
&\le \mathbb{E}\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}}\Big[Z_i^{(c)}(t)\Big(1-\delta_i^{(c)}\Big)\Big(Q_i^{(c)}(t)+R_i^{(c)}(t)\Big)\\
&\qquad\qquad - \Big(Z_i^{(c)}(t)+Q_i^{(c)}(t)+R_i^{(c)}(t)\Big)1_i\\
&\qquad\qquad \times (t)\Big(D_{r,i}^{\hat{(c)}}(t)+\hat{D}_{l,i}^{(c)}(t)\Big)\Big|\overrightarrow{U}(t)\Big]\\
&\quad -\mathbb{E}\Bigg[\sum_{i\in\mathcal{N}}\tilde{E}_i(t)\Bigg[1_i(t)\sum_{c\in\mathcal{C}}\Big(p_{l,i}^{\hat{(c)}}(t)+p_{tx,i}^{\hat{(c)}}(t)\Big)-\hat{e}_i(t)\Bigg]\Bigg|\overrightarrow{U}(t)\Bigg]\\
&\quad + B + V\cdot\mathbb{E}\Big[\hat{P}(t)|\overrightarrow{U}(t)\Big]\\
&\le B + V\cdot\overline{P}^{\mathrm{opt}}.
\end{aligned}\tag{42}$$

The second inequality is due to the optimal solution for P1′, $\mathbb{E}[1_i(t)\sum_{c\in\mathcal{C}}(p_{l,i}^{\hat{(c)}}(t)+p_{tx,i}^{\hat{(c)}}(t))-\hat{e}_i(t)]=0$, the stability of data queue and $\hat{\overline{P}} \le \overline{P}^{\mathrm{opt}}$. Then, taking expectation on the both sides of the inequalities and summing it for $t=0,\ldots,T-1$, we have

$$\begin{aligned}
\sum_{t=0}^{T-1} V\cdot\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big] &\le T\cdot B + T\cdot V\cdot\overline{P}^{\mathrm{opt}} - \mathbb{E}\{L(T)\}\\
&\quad + \mathbb{E}\{L(0)\}\\
&\le T\cdot B + T\cdot V\cdot\overline{P}^{\mathrm{opt}} + \mathbb{E}\{L(0)\}.
\end{aligned}\tag{43}$$

The second inequality is due to the non-negative property of Lyapunov functions. Dividing both sides by $K\cdot T$ and let T go to infinity, and according to the initial system state is finite, we have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1} V\mathbb{E}\Big[P(t)|\overrightarrow{U}(t)\Big] \le \overline{P}^{\mathrm{opt}} + \frac{B}{V}.\tag{44}$$

This completes the proof of Theorem 2: $\overline{P}^{*} \le \overline{P}^{\mathrm{opt}} + (B/V)$.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] Y. Wu *et al.*, “Secrecy-driven resource management for vehicular computation offloading networks,” *IEEE Netw.*, vol. 32, no. 3, pp. 84–91, May/Jun. 2018.

[3] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.

[4] “Mobile-edge computing-introductory,” Sophia Antipolis, France, Eur. Telecommun. Stand. Inst., White Paper, Sep. 2014.

[5] 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.

[6] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: The role of MEC in the Internet of Things,” *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[7] A. Ahmed and E. Ahmed, “A survey on mobile edge computing,” in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[8] Y. Wu, L. P. Qian, J. Zheng, H. Zhou, and X. S. Shen, “Green-oriented traffic offloading through dual connectivity in future heterogeneous small cell networks,” *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 140–147, May 2018.

[9] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[10] D. Zhang *et al.*, “Energy-harvesting-aided spectrum sensing and data transmission in heterogeneous cognitive radio sensor network,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 831–843, Jan. 2017.

[11] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, “Power-delay tradeoff in multi-user mobile-edge computing systems,” in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[12] D. Zhang *et al.*, “Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3552–3565, Dec. 2016.

[13] D. Zhang, R. Shen, J. Ren, and Y. Zhang, “Delay-optimal proactive service framework for block-stream as a service,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 598–601, Aug. 2018.

[14] Y. Huang, S. Mao, and R. M. Nelms, “Adaptive electricity scheduling in microgrids,” *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 270–281, Jan. 2014.

[15] M. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan Claypool, 2010.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *J. Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[17] W. Zhang *et al.*, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[18] T. B. Burd and R. W. Brodersen, “Processor design for portable systems,” *J. VLSI Signal Process. Syst.*, vol. 13, nos. 2–3, pp. 203–221, Aug. 1996.

[19] Z. Wang, V. Aggarwal, and X. Wang, “Joint energy-bandwidth allocation in multiple broadcast channels with energy harvesting,” *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3842–3855, Oct. 2015.

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

[21] S. Sun, M. Dong, and B. Liang, “Distributed real-time power balancing in renewable-integrated power grids with storage and flexible loads,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2337–2349, Sep. 2016.

[22] H. Wang and A. Banerjee. *Online Alternating Direction Method*. Accessed: Jan. 1, 2018. [Online]. Available: http://arxiv.org/abs/1306.3721

[23] 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, Aug. 2017.

[24] X. Peng *et al.*, “BOAT: A block-streaming app execution scheme for lightweight IoT devices,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1816–1829, Jun. 2018.

[25] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[26] D. Huang, P. Wang, and D. Niyato, “A dynamic offloading algorithm for mobile computing,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[27] C. You and K. Huang, “Multiuser resource allocation for mobile-edge computation offloading,” in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

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

[29] X. Lyu *et al.*, “Optimal schedule of mobile edge computing for Internet of Things using partial information,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov. 2017.

[30] W. Labidi, M. Sarkiss, and M. Kamoun, “Joint multi-user resource scheduling and computation offloading in small cell networks,” in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Oct. 2015, pp. 794–801.

[31] 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.

[32] J. Xu, L. Chen, and S. Ren, “Online learning for offloading and autoscaling in energy harvesting mobile edge computing,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[33] D. Zhang *et al.*, “Two time-scale resource management for green Internet of Things networks,” *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2842766.

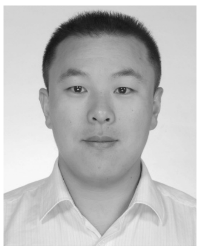
[34] G. Piro *et al.*, "HetNets powered by renewable energy sources: Sustainable next-generation cellular networks," *IEEE Internet Comput.*, vol. 17, no. 1, pp. 32–39, Jan./Feb. 2013.

[35] Y. Mao, Y. Luo, J. Zhang, and K. B. Letaief, "Energy harvesting small cell networks: Feasibility, deployment, and operation," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 94–101, Jun. 2015.

[36] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[37] S. Mao, S. Leng, K. Yang, Q. Zhao, and M. Liu, "Energy efficiency and delay tradeoff in multi-user wireless powered mobile-edge computing systems," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[38] L. Chen, S. Zhou, and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.



Guanglin Zhang (M'14) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012.

From 2013 to 2014, he was a Post-Doctoral Research Associate with the Institute of Network Coding, Chinese University of Hong Kong, Hong Kong. He is currently a Professor and the Department Chair with the Department of Communication Engineering, Donghua University, Shanghai. His current research interests include capacity scaling of wireless networks, vehicular networks, smart micro-grid, and mobile edge computing.

Dr. Zhang serves as a Technical Program Committee Member for IEEE Globecom 2016-2017, IEEE ICC 2014, 2015, 2017, IEEE VTC2017-Fall, IEEE/CIC ICCC 2014, WCSP 2014, APCC 2013, and WASA 2012. He serves as the Local Arrangement Chair of ACM TURC 2017, and the Vice TPC Co-Chair of ACM TURC 2018. He serves as an Editor on the Editorial Board of China Communications and the *Journal of Communications and Information Networks*. He is an associate editor of IEEE ACCESS.

Yan Chen received the B.S. degree in communication engineering from Donghua University, Shanghai, China, in 2017, where she is currently pursuing the master's degree in information and communication engineering.

Her current research interests include the energy management in mobile edge computing and distributed optimization.

Zhirong Shen received the B.S. degree in communication engineering from Donghua University, Shanghai, China, in 2017, where he is currently pursuing the Ph.D. degree in information and communication intelligence system.

His current research interests include the offloading delay in MEC, the energy scheduling and the co-generation in microgrids.

Lin Wang (M'14) received the Ph.D. degree in control theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2009.

She joined the Department of Automation, Shanghai Jiao Tong University (SJTU), Shanghai. She was a Visiting Scholar with the Division of Optimization and Systems Theory, Royal Institute of Technology (KTH), Stockholm, Sweden, in 2010, for six months and from 2011 to 2012. She is currently an Associate Professor with SJTU. Her current research interests include multiagent systems, adaptive complex networks, and coordination of multiple manipulators.