Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems

Yuyi Mao, Jun Zhang, S.H. Song, and Khaled B. Letaief, *Fellow, IEEE*

Abstract

Mobile-edge computing (MEC) has recently emerged as a prominent technology to liberate mobile devices from computationally intensive workloads, by offloading them to the proximate MEC server. To make offloading effective, the radio and computational resources need to be dynamically managed, to cope with the time-varying computation demands and wireless fading channels. In this paper, we develop an online joint radio and computational resource management algorithm for multi-user MEC systems, with the objective as minimizing the long-term average weighted sum power consumption of the mobile devices and the MEC server, subject to a task buffer stability constraint. Specifically, at each time slot, the optimal CPU-cycle frequencies of the mobile devices are obtained in closed forms, and the optimal transmit power and bandwidth allocation for computation offloading are determined with the *Gauss-Seidel method*; while for the MEC server, both the optimal frequencies of the CPU cores and the optimal MEC server scheduling decision are derived in closed forms. Besides, a delay-improved mechanism is proposed to reduce the execution delay. Rigorous performance analysis is conducted for the proposed algorithm and its delay-improved version, indicating that the weighted sum power consumption and execution delay obey an $\left[O\left(1/V\right), O\left(V\right)\right]$ tradeoff with V as a control parameter. Simulation results are provided to validate the theoretical analysis and demonstrate the impacts of various parameters.

Index Terms

Mobile-edge computing, dynamic voltage and frequency scaling, radio and computational resource management, Lyapunov optimization.

The authors are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: {ymaoac, eejzhang, eeshsong, eekhaled}@ust.hk). Khaled B. Letaief is also with Hamad bin Khalifa University, Doha, Qatar (e-mail: kletaief@hbku.edu.qa).

Part of this work was presented at IEEE Global Communications Conference (GLOBECOM), Washington, DC, Dec. 2016 [1].

I. Introduction

The increasing popularity of smart mobile devices is driving the development of computation-intensive mobile applications with advanced features, e.g., interactive online gaming, gesture and face recognition, voice control, as well as 3D modeling. This poses more stringent requirements on the quality of computation experience, which cannot be easily satisfied by mobile devices due to their limited resources, e.g., the processing speed, memory size, and battery energy. As a result, new solutions to handle the explosive computation demands and the ever-increasing computation quality requirements are emerging [2]. *Mobile-edge computing* (MEC) is such a promising technology to release the tension between the computation-intensive applications and the resource-limited mobile devices [3]–[5]. Different from conventional cloud computing systems, which rely on remote public clouds that will induce long latency due to data exchange, MEC offers computation capability within the radio access network. Therefore, by offloading the computation tasks from the mobile devices to the MEC servers, the quality of computation experience, including energy consumption and execution latency, can be greatly improved [6]–[9].

A. Related Works

Computation offloading for cloud computing systems has attracted significant attention from computer science and communications research communities in recent years. In order to prolong the battery lifetime and improve the computation performance, various code offloading frameworks, e.g., MAUI [10] and ThinkAir [11], were proposed. Nevertheless, the efficiency of computation offloading for MEC highly depends on the wireless channel condition, as computation offloading requires effective wireless data transmission between mobile devices and MEC servers. Therefore, computation offloading policies that take the wireless channel condition into consideration have been extensively studied most recently [12]–[26]. In [12], for tasks with a strict execution deadline, the local execution energy consumption was minimized by adopting *dynamic voltage and frequency scaling* (DVFS) techniques, and the energy consumption for computation offloading was optimized via data transmission scheduling. In [13], a joint optimization of communication and computational resource allocation for femto-cloud computing systems was proposed, where the cloud server is formed by a set of femto access points. This study was extended to the *cloud radio access networks* (C-RANs) with mobile cloud computing capability in [14]. Besides, resource allocation policies were proposed for MEC systems based on *time division* *multiple access* (TDMA) and *orthogonal frequency-division multiple access* (OFDMA) in [15], [16], while game-theoretic decentralized computation offloading algorithms were proposed for multi-user MEC systems in [17] and [18] for single- and multi-channel wireless environments, respectively. Moreover, dynamic computation offloading policies have been developed for MEC systems powered by energy harvesting [19] and wireless power transfer [20], for applications where replacing/recharing the device batteries is costly and difficult. Most recently, a novel task offloading framework based on network-assisted *device-to-device* (D2D) communications was proposed in [21], which enables resource sharing among the mobile users.

However, there are some limitations in the commonly adopted assumptions in [12]–[16], [19]–[21]: It is typically assumed that the computation tasks have strict delay requirements, and no new task will be generated before the old tasks are completed/abandoned. Such assumptions make the computation offloading design more tractable, as only short-term performance, e.g., the performance for executing a single task for each mobile device, needs to be considered and thus the associated optimization problems are typically deterministic. Nevertheless, they may be impractical for applications that can tolerate a certain period of execution latency, e.g., multi-media streaming and file backup. For such types of applications, the long-term system performance is more relevant, and stochastic task models should be adopted. In particular, the coupling among the randomly arrived tasks cannot be ignored, and stochastic computation offloading policies should be developed [22]–[26]. In order to minimize the long-term average energy consumption, a stochastic control algorithm was proposed in [22], which determines the offloaded software components in an application. In [23], a delay-optimal stochastic task scheduling policy for single-user MEC systems was proposed based on the Markov decision process. The energy-delay tradeoff in single-user MEC systems with a multi-core mobile device and heterogeneous types of mobile applications were investigated in [24] and [25], respectively. For MEC systems with multiple devices, the optimal design becomes more challenging compared to single-user MEC systems, as the computational resource for task execution and the radio resource for computation offloading are shared by multiple mobile devices. In other words, the optimal system operations in multi-user MEC systems are not only temporally correlated due to the random computation task arrivals, but also spatially coupled due to the competition among multiple devices. Moreover, the freedom of parallel local and remote processing makes the mobile execution strategy, computation offloading policy, and the operations at the MEC server interdependent. Consequently, intelligent joint allocation of the radio and computational resource should be considered to maximize the benefits of MEC. An initial investigation for multi-user MEC systems with delay-tolerant applications was conducted in [26], which, however, only focused on computational resource scheduling and failed to address radio resource management.

B. Contributions

In this paper, we investigate stochastic joint radio and computational resource management for multi-user MEC systems. Our major contributions are summarized as follows:

- We consider a general MEC system with multiple mobile devices and a physically proximate MEC server with *frequency division multiple access* (FDMA). The MEC server has limited computation capability, which generalizes our previous work in [1], where the MEC server is assumed to be computationally powerful with unlimited computational resources.
- The average weighted sum power consumption of the mobile devices and the MEC server is adopted as the performance metric, which is able to address the cost of power consumption at different nodes in MEC systems. The available radio and computational resources are jointly managed to optimize the MEC system, including the CPU-cycle frequencies for the mobile and server CPUs, the transmit power and bandwidth allocation for computation offloading, as well as the task scheduling decision at the MEC server[1]. This is a critical but highly non-trivial design consideration for multi-user MEC systems, since the optimal system operations are temporally and spatially correlated due to the stochastic computation task arrivals and the competition among multiple devices for the available resources, respectively.
- An average weighted sum power consumption minimization problem subject to a task buffer stability constraint is formulated, assuming causal *side information* (SI) of the task arrival and wireless channel processes. This is a very challenging stochastic optimization problem, which involves a large amount of SI as well as decision variables. A low-complexity online algorithm is then proposed based on Lyapunov optimization. In each time slot, the system operation is determined by solving a deterministic problem, where the CPU-cycle frequencies and the MEC server scheduling decision are obtained in closed forms, while the transmit power and bandwidth allocation are obtained through an efficient Gauss-Seidel method. Besides, a delay-improved mechanism is designed for the proposed algorithm.

[1] In the conference version of this paper [1], only the CPU-cycle frequencies for the local CPUs, the transmit power and bandwidth allocation for computation offloading were optimized since the MEC server is assumed to have unlimited computational resources. Besides, the power consumption of the MEC server was viewed as a constant and not included in the optimization in [1].

- Performance analysis is conducted for the proposed algorithm and its delay-improved version, which not only shows their capability in achieving asymptotic optimality, but also explicitly characterizes the tradeoff between the weighted sum power consumption and the execution delay. Simulation results corroborate the theoretical analysis and show that the proposed algorithms are able to balance the weighted sum power consumption and execution delay performance. In addition, the impacts of various parameters are revealed, which demonstrate the necessity of a joint consideration on radio and computational resource management for multi-user MEC systems, and offer valuable guidelines for real deployment.

C. *Organization*

The organization of this paper is as follows. We introduce the system model and formulate the average weighted sum power consumption minimization problem in Section II and Section III, respectively. An online joint radio and computational resource management algorithm as well as a delay-improved mechanism are developed in Section IV. We conduct performance analysis for the proposed algorithms in Section V. Simulation results will be shown in Section VI, and we will conclude this paper in Section VII.

II. System Model

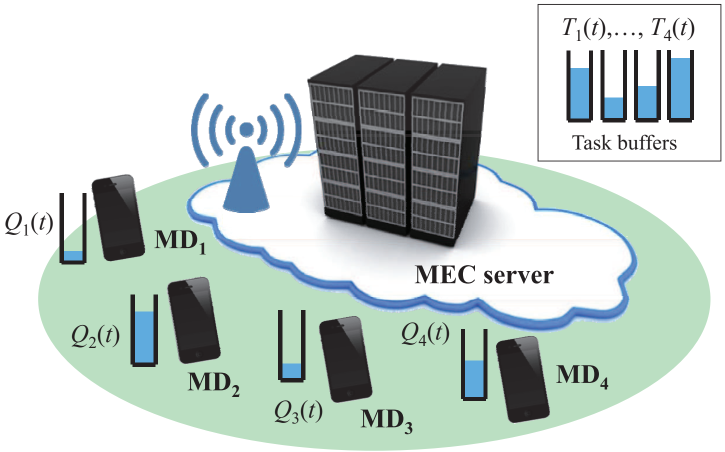

Fig. 1. A mobile-edge computing system with four mobile devices (MDs). The queue lengths of the task buffer at the ith mobile device, and the task buffer maintained by the MEC server for this device at the beginning of the tth time slot, are denoted as $Q_i(t)$ and $T_i(t)$, respectively.

We consider a mobile-edge computing (MEC) system as shown in Fig. 1, where N single-core mobile devices running computation-intensive applications are assisted by an MEC server.

This corresponds to the scenarios where the mobile devices have relatively weak computation capability and the MEC is of supreme need, such as the *Internet of Things* (IoT) applications and the *wireless sensor networks* (WSNs) for surveillance [27], [28]. However, our proposed algorithms can be adapted for multi-core mobile devices with minor modifications. The MEC server could be a small data center installed at a wireless *access point* (AP) deployed by the telecom operator. Therefore, it can be accessed by the mobile devices through wireless channels, and will execute the computation tasks on behalf of the mobile devices [5], [6], [12]. Besides, with the readily available wireless channel state information at the wireless AP, joint radio and computational resource management can be realized in MEC systems, which is an innovative feature that cannot be supported in traditional cloud computing systems. By offloading part of the computation tasks to the MEC server, the mobile devices could not only enjoy a higher level of quality of computation experience, but also reduce the battery energy consumption [6]–[9].

The available system bandwidth is ω Hz, which is shared by the mobile devices using FDMA, and the noise power spectral density at the receiver of the MEC server is denoted as N_0. Time is slotted and the time slot length is τ. For convenience, we denote the index sets of the mobile devices and the time slots as $\mathcal{N} \triangleq \{1, \cdots, N\}$ and $\mathcal{T} \triangleq \{0, 1, \cdots\}$, respectively. For ease of reference, we list the key notations of our system model in Table I.

A. Computation Task And Task Queueing Models

We assume that the mobile devices are running independent and fine-grained tasks [16], [25], [26], [29]: At the beginning of the tth time slot, $A_i(t)$ (bits) of computation tasks arrive at the ith mobile device, which can be processed starting from the $(t+1)$th time slot. Without loss of generality, we assume the $A_i(t)$'s in different time slots are independent and identically distributed (i.i.d.) within $[A_{i,\min}, A_{i,\max}]$ with $\mathbb{E}[A_i(t)] = \lambda_i, i \in \mathcal{N}$.

In each time slot, part of the computation tasks at the ith mobile device, denoted as $D_{l,i}(t)$, will be executed at the local CPU, while $D_{r,i}(t)$ bits of the computation tasks will be offloaded to the MEC server. The arrived but not yet executed (or offloaded) tasks will be queued in the task buffer at each mobile device with sufficiently large capacity [22], [24]–[26], and the queue lengths of the task buffers at the beginning of the tth time slot are denoted as $\mathbf{Q}(t) \triangleq [Q_1(t), \cdots, Q_N(t)]$, where $Q_i(t)$ evolves according to the following equation:

$$Q_i(t+1) = \max\{Q_i(t) - D_{\Sigma,i}(t), 0\} + A_i(t), t \in \mathcal{T}. \tag{1}$$

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	**Description**
ω	System bandwidth
$\mathcal{T}$ ($\mathcal{N}$)	Index set of the time slots (mobile devices)
$A_i\left(t\right)$	Amount of tasks arrived at the ith mobile device at the beginning time slot t
$Q_i\left(t\right)$	Queue length of the task buffer at the ith mobile device at the beginning of time slot t
$T_i\left(t\right)$	Queue length of the task buffer maintained by the MEC server for the ith mobile device at the beginning of time slot t
$D_{l,i}\left(t\right)$	Amount of tasks executed locally at the ith mobile device in time slot t
$D_{r,i}\left(t\right)$	Amount of tasks offloaded to the MEC server in time slot t by the ith mobile device
$D_{s,i}\left(t\right)$	Amount of tasks from the ith mobile device executed by the MEC server in time slot t
$f_i\left(t\right)$	CPU-cycle frequency of the ith mobile device in time slot t
$p_{l,i}\left(t\right)$	Power consumption of the CPU at the ith mobile device in time slot t
$p_{\mathrm{tx},i}\left(t\right)$	Transmit power of the ith mobile device in time slot t
$f_{C,m}\left(t\right)$	CPU-cycle frequency of the mth CPU core at the MEC server in time slot t
$p_{\mathrm{ser}}\left(t\right)$	Power consumption of the MEC server in time slot t
$\alpha_i\left(t\right)$	Proportion of bandwidth allocated for the ith mobile device in time slot t
$\Gamma_i\left(t\right)$	Channel power gain from the ith mobile device to the MEC server in time slot t

In (1), $D_{\Sigma,i}\left(t\right) \triangleq D_{l,i}\left(t\right) + D_{r,i}\left(t\right)$ is the amount of tasks departing from the task buffer at the ith mobile device in the tth time slot.

The MEC server maintains a task buffer for each mobile device to store the tasks that have been offloaded but not yet executed by the server, which is assumed to with sufficiently large capacity similar to the one at the mobile side. Denote the queue lengths of the task buffers at the MEC server at the beginning of the tth time slot as $\mathbf{T}\left(t\right) \triangleq \left[T_1\left(t\right), \cdots, T_N\left(t\right)\right]$. We further denote the task scheduling decision of the MEC server at the tth time slot as $D_{s,n}\left(t\right), n \in \mathcal{N}$, where $D_{s,n}\left(t\right)$ is the amount of tasks from the nth mobile device executed by the MEC server in time slot t. Therefore, $T_i\left(t\right)$ evolves according to the following equation:

$$T_i\left(t+1\right) = \max\{T_i\left(t\right) - D_{s,i}\left(t\right), 0\} + \min\{\max\{Q_i\left(t\right) - D_{l,i}\left(t\right), 0\}, D_{r,i}\left(t\right)\}, t \in \mathcal{T}, \quad (2)$$

which indicates that only the tasks that have not been executed locally at the mobile devices will be stored in the task buffers at the MEC server. It is worthwhile to note that the departure function $D_{\Sigma,i}\left(t\right)$ may be larger than the amount of tasks in the corresponding local task buffer $Q_i\left(t\right)$, depending on the decided system operation. In this case, the excessive communication rates are allocated for transmitting dummy task inputs, which are not necessary to be processed by the MEC server. Without loss of generality, we assume the tasks buffers are empty initially, i.e., $Q_i\left(0\right) = T_i\left(0\right) = 0, i \in \mathcal{N}$.

B. Local Execution Model

In order to process one bit of computation task input from the ith mobile device, L_i CPU cycles will be needed, which depends on the types of applications and can be obtained by off-line measurements [30]. Denote the CPU-cycle frequency of the ith mobile device in the tth time slot as $f_i\left(t\right)$, which cannot exceed its maximum value $f_{i,\max}$. Thus, $D_{l,i}\left(t\right)$ can be expressed as

$$D_{l,i}\left(t\right) = \tau f_i\left(t\right) L_i^{-1}. \tag{3}$$

According to circuit theories, the CPU power is dominated by the dynamic power, which originates from the toggling activities of the logic gates inside the CPU, and proportional to $v_{\mathrm{cir}}^2 f_c$ in CMOS circuits, where v_{cir} and f_c are the circuit voltage and CPU-cycle frequency, respectively [31], [32]. Besides, when operating at the low voltage limits, the CPU-cycle frequency is approximately linear to the chip voltage [32]. Therefore, the power consumption for local execution at the ith mobile device is given by[2]

$$p_{l,i}\left(t\right) = \kappa_{\mathrm{mob},i} f_i^3\left(t\right), \tag{4}$$

where $\kappa_{\mathrm{mob},i}$ is the effective switched capacitance of the CPU at the ith mobile device, and it is related to the chip architecture [32].

C. MEC Server Execution Model

1) Computation Offloading: To offload the computation tasks for MEC server execution, the input bits of the tasks need to be delivered to the MEC server. We assume the wireless channels between the mobile devices and the MEC server are i.i.d. frequency-flat block fading. Denote the small-scale fading channel power gain from the ith mobile device to the MEC server in the tth time slot as $\gamma_i\left(t\right)$, which is assumed to have a bounded mean value, i.e., $\mathbb{E}\left[\gamma_i\left(t\right)\right] \triangleq \overline{\gamma_i} < \infty$. Thus, the channel power gain from the ith mobile device to the MEC server can be represented

[2] In order to apply the proposed algorithms in scenarios where multi-core CPUs are available at the devices, $D_{l,i}\left(t\right)$ and $p_{l,i}\left(t\right)$ should be modified as $\tau L_i^{-1} \cdot \sum_{z=1}^{Z_i} f_{i,z}\left(t\right)$ and $\sum_{z=1}^{Z_i} \kappa_{\mathrm{mob},i,z} f_{i,z}^3\left(t\right)$, respectively. Here, $Z_i \geq 1$ is the number of CPU cores at the ith mobile device, $f_{i,z}\left(t\right)$, $\kappa_{\mathrm{mob},i,z}$ and $f_{i,z,\max}$ denote the CPU-cycle frequency, effective switched capacitance and maximum CPU-cycle frequency of the zth CPU core at the ith mobile device, respectively. Then, $\left\{f_{i,z}\left(t\right)\right\}$'s could be determined for each mobile device under constraints $0 \leq f_{i,z}\left(t\right) \leq f_{i,z,\max}, z = 1, \cdots, Z_i, i \in \mathcal{N}$ with the proposed algorithms.

by $\Gamma_i(t) = \gamma_i(t) g_0 (d_0/d_i)^\theta$, where g_0 is the path-loss constant, θ is the path-loss exponent, d_0 is the reference distance, and d_i is the distance from the ith mobile device to the MEC server. Since FDMA is utilized, according to the Shannon-Hartley formula [33], the amount of computation tasks offloaded from the ith mobile device in time slot t is given by

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

where $p_{\mathrm{tx},i}(t)$ is the transmit power with the maximum value of $p_{i,\max}$, and $\alpha_i(t)$ is the proportion of bandwidth allocated to the ith mobile device. Denote $\boldsymbol{\alpha}(t) \triangleq [\alpha_1(t), \cdots, \alpha_N(t)]$ as the bandwidth allocation vector, which should be chosen from the feasible set $\mathcal{A}$, i.e., $\boldsymbol{\alpha}(t) \in \mathcal{A} \triangleq \{\boldsymbol{\alpha} \in \mathbb{R}_+^N \big| \sum_{i\in\mathcal{N}} \alpha_i \leq 1\}$ [34].

2) MEC Server Scheduling: The MEC server is equipped with an M-core CPU, where the CPU cores could be heterogeneous and the set of CPU cores is denoted as $\mathcal{M} \triangleq \{1, \cdots, M\}$. Denote the CPU-cycle frequency of the mth CPU core in the tth time slot as $f_{C,m}(t)$, which should be less than its maximum value $f_{C_m,\max}$. Thus, the power consumption of the CPU cores at the MEC server can be expressed as

$$p_{\mathrm{ser}}(t) = \sum_{m\in\mathcal{M}} \kappa_{\mathrm{ser},m} f_{C,m}^3(t), \tag{6}$$

where $\kappa_{\mathrm{ser},m}$ is the effective switched capacitance of the mth CPU core at the MEC server.

The CPU cycles offered by the server CPU can be allocated for the computation tasks offloaded from different mobile devices, i.e., the MEC server scheduling decision should satisfy the following constraint:

$$\sum_{n\in\mathcal{N}} D_{s,n}(t) L_n \leq \sum_{m\in\mathcal{M}} f_{C,m}(t)\tau, t \in \mathcal{T}, \tag{7}$$

which means the number of CPU cycles needed for completing $\mathbf{D}_s(t) \triangleq [D_{s,1}(t), \cdots, D_{s,N}(t)]$ should be no larger than the available CPU cycles at the MEC server.

III. Problem Formulation

In this section, we will first introduce the performance metrics, namely, the average weighted sum power consumption of the MEC system and the average sum queue length of the task buffers. An average weighted sum power consumption minimization problem with a task buffer stability constraint will then be formulated.

A. Performance Metrics

We focus on the power consumption of the task execution processes in the local and server CPUs, as well as the transmit power for computation offloading. Energy consumed for other purposes, e.g., powering the screens of the mobile devices and supporting the basic operations in the MEC system, is ignored for simplicity. Therefore, we adopt the average weighted sum power consumption of different entities in the MEC system as the performance metric, which is defined as follows:

$$\overline{P}_{\Sigma} \triangleq \lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i \in \mathcal{N}} w_i \left(p_{\mathrm{tx},i}(t) + p_{l,i}(t) \right) + w_{N+1} p_{\mathrm{ser}}(t) \right], \tag{8}$$

where $w_i \geq 0, i \in \mathcal{N}$ is the weight of the power consumption at the ith mobile device and $w_{N+1} \geq 0$ is the weight of the power consumption at the MEC server. These parameters can be adjusted to address the cost of power consumption at different nodes in the MEC system, as well as to balance the power consumption of the mobile devices and the MEC server [35], which are assumed to be constants throughout this paper. For convenience, we denote $P_{\Sigma}(t) \triangleq \sum_{i \in \mathcal{N}} w_i \left(p_{\mathrm{tx},i}(t) + p_{l,i}(t) \right) + w_{N+1} p_{\mathrm{ser}}(t)$, i.e., $\overline{P}_{\Sigma} = \lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[P_{\Sigma}(t) \right]$.

According to *Little's Law* [36], the average execution delay experienced by each mobile device is proportional to the average number of its tasks waiting in the MEC system, which is the sum queue length of the task buffers at the device and server sides. Thus, the average sum queue length of the task buffers for each mobile device is used as a measurement of the execution delay, which can be written as

$$\overline{q}_{\Sigma,i} = \lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[Q_i(t) + T_i(t) \right], i \in \mathcal{N}. \tag{9}$$

B. Average Weighted Sum Power Consumption Minimization

We denote the system operation at the tth time slot as $\mathbf{X}(t) \triangleq [\mathbf{f}(t), \mathbf{p}_{\mathrm{tx}}(t), \boldsymbol{\alpha}(t), \mathbf{f}_C(t), \mathbf{D}_s(t)]$, where $\mathbf{f}(t) \triangleq [f_1(t), \cdots, f_N(t)]$, $\mathbf{p}_{\mathrm{tx}}(t) \triangleq [p_{\mathrm{tx},1}(t), \cdots, p_{\mathrm{tx},N}(t)]$ and $\mathbf{f}_C(t) \triangleq [f_{C,1}(t), \cdots, f_{C,M}(t)]$. Therefore, the average weighted sum power consumption minimization problem can be formulated in $\mathbf{P_1}$, where (10) is the bandwidth allocation constraint, while (11) denotes the CPU-cycle frequency and the transmit power constraints for the mobile devices. (12) is the CPU-cycle frequencies constraint for the M CPU cores at the MEC server. The MEC server scheduling constraint is imposed by (13). (14) enforces the task buffers to be mean rate stable [37], which guarantees that all the arrived computation tasks can be completed with finite delay.

$$\begin{aligned} \mathbf{P_1} : \min_{\{\mathbf{X}(t)\}} \quad & \overline{P}_{\Sigma} \\ \text{s.t.} \quad & \boldsymbol{\alpha}(t) \in \mathcal{A}, t \in \mathcal{T} & (10) \\ & 0 \leq f_i(t) \leq f_{i,\max}, 0 \leq p_{\text{tx},i}(t) \leq p_{i,\max}, i \in \mathcal{N}, t \in \mathcal{T} & (11) \\ & 0 \leq f_{C,m}(t) \leq f_{C_m,\max}, m \in \mathcal{M}, t \in \mathcal{T} & (12) \\ & \sum_{n \in \mathcal{N}} D_{s,n}(t) L_n \leq \sum_{m \in \mathcal{M}} f_{C,m}(t) \tau, D_{s,i}(t) \geq 0, i \in \mathcal{N}, t \in \mathcal{T} & (13) \\ & \lim_{T \to +\infty} \frac{\mathbb{E}[|Q_i(T)|]}{T} = 0, \lim_{T \to +\infty} \frac{\mathbb{E}[|T_i(T)|]}{T} = 0, i \in \mathcal{N}. & (14) \end{aligned}$$

Remark *1:* It is not difficult to identify that $\mathbf{P_1}$ is a stochastic optimization problem, for which, the operations at both the mobile device side (including the CPU-cycle frequencies for the local CPUs, as well as the transmit power and bandwidth allocation for computation offloading) and the MEC server side (including the MEC server scheduling and the CPU-cycle frequencies for the multiple CPU cores) need to be determined at each time slot. This is a highly challenging problem with a large amount of SI (including the channel and task buffer state information) to be handled and a large number of variables to be determined. Also, the optimal decisions are temporally correlated due to the randomly arrived tasks. Besides, an efficient resource management policy for the mobile devices and the MEC server is critical since offloading the computation tasks in either an over-conservative or an over-aggressive manner will result in ineffective use of the available computational resources. Moreover, the spatial coupling of the bandwidth allocation among different mobile devices poses an additional challenge for the radio resource management, which is also interdependent with the computational resource management. Therefore, a joint optimization of the radio and computational resource allocation is essential.

Instead of solving $\mathbf{P_1}$ directly, we consider its modified version, denoted as $\mathbf{P_2}$, which is obtained by replacing $\mathcal{A}$ in (10) by $\tilde{\mathcal{A}}$ with $\tilde{\mathcal{A}} \triangleq \{\boldsymbol{\alpha} \in \mathbb{R}_+^N | \sum_{i \in \mathcal{N}} \alpha_i \leq 1, \alpha_i \geq \epsilon_A, i \in \mathcal{N}\}$, $\epsilon_A \in (0, 1/N)$. Thus, the task departure function of computation offloading, i.e., $D_{r,i}(t)$, is continuous and differentiable with respect to $\boldsymbol{\alpha}(t) \in \tilde{\mathcal{A}}$, which helps to develop an efficient asymptotically optimal online algorithm for $\mathbf{P_2}$ that is also feasible for $\mathbf{P_1}$. Besides, although the optimal value of $\mathbf{P_2}$ is larger than that of $\mathbf{P_1}$, they can be made arbitrarily close by setting ϵ_A to be sufficiently small. As a result, we will focus on $\mathbf{P_2}$ in the remainder of this paper.

IV. Online Joint Radio and Computational Resource Management Algorithm

In this section, we will propose an online joint radio and computational resource management algorithm to solve $\mathbf{P_2}$ based on Lyapunov optimization [37]. With the assistance of the Lyapunov optimization framework, we are able to resolve this challenging stochastic optimization problem by solving a deterministic per-time slot problem at each time slot, for which, the optimal solution can be obtained with low complexity. Besides, a delay-improved mechanism will be designed for the proposed Lyapunov optimization-based algorithm. In the next section, we will show the proposed algorithm and its delay-improved version are capable of achieving asymptotic optimality and reveal the power-delay tradeoff in multi-user MEC systems.

A. The Lyapunov Optimization-Based Online Algorithm

To present the algorithm, we first define the Lyapunov function as

$$L\left(\mathbf{\Theta}\left(t\right)\right) = \frac{1}{2}\sum_{i\in\mathcal{N}}\left[Q_i^2\left(t\right) + T_i^2\left(t\right)\right], \tag{15}$$

where $\mathbf{\Theta}\left(t\right) \triangleq \left[\mathbf{Q}\left(t\right), \mathbf{T}\left(t\right)\right]$. Thus, the conditional Lyapunov drift can be written as

$$\Delta\left(\mathbf{\Theta}\left(t\right)\right) = \mathbb{E}\left[L\left(\mathbf{\Theta}\left(t+1\right)\right) - L\left(\mathbf{\Theta}\left(t\right)\right)\left|\mathbf{\Theta}\left(t\right)\right.\right]. \tag{16}$$

Accordingly, the Lyapunov drift-plus-penalty function can be expressed as

$$\Delta_V\left(\mathbf{\Theta}\left(t\right)\right) = \Delta\left(\mathbf{\Theta}\left(t\right)\right) + V\cdot\mathbb{E}\left[P_\Sigma\left(t\right)\left|\mathbf{\Theta}\left(t\right)\right.\right], \tag{17}$$

where $V \in (0, +\infty)$ ($\mathrm{bits}^2 \cdot \mathrm{W}^{-1}$) is a control parameter in the proposed algorithm. We first find an upper bound of $\Delta_V\left(\mathbf{\Theta}\left(t\right)\right)$ under any feasible $\mathbf{X}\left(t\right)$, as specified in Lemma 1.

***Lemma** 1:* For arbitrary $\mathbf{X}\left(t\right)$ such that $f_i\left(t\right) \in [0, f_{i,\max}]$, $p_{\mathrm{tx},i}\left(t\right) \in [0, p_{i,\max}]$, $D_{s,i}\left(t\right) \geq 0, i \in \mathcal{N}$, $f_{C,m}\left(t\right) \in [0, f_{C_m,\max}]$, $m \in \mathcal{M}$, $\sum_{i\in\mathcal{N}} D_{s,i}\left(t\right) L_i \leq \sum_{m\in\mathcal{M}} f_{C,m}\left(t\right)\tau$, and $\boldsymbol{\alpha}\left(t\right) \in \tilde{\mathcal{A}}$, $\Delta_V\left(t\right)$ is upper bounded, i.e.,

$$\begin{aligned}\Delta_V\left(\mathbf{\Theta}\left(t\right)\right) \leq C &- \mathbb{E}\left[\sum_{i\in\mathcal{N}} Q_i\left(t\right)\left(D_{\Sigma,i}\left(t\right) - A_i\left(t\right)\right)\left|\mathbf{\Theta}\left(t\right)\right.\right] \\ &- \mathbb{E}\left[\sum_{i\in\mathcal{N}} T_i\left(t\right)\left(D_{s,i}\left(t\right) - D_{r,i}\left(t\right)\right)\left|\mathbf{\Theta}\left(t\right)\right.\right] + V\cdot\mathbb{E}\left[P_\Sigma\left(t\right)\left|\mathbf{\Theta}\left(t\right)\right.\right],\end{aligned} \tag{18}$$

where C is a constant.

Proof: Please refer to Appendix A. ■

The main idea of the proposed online joint radio and computational resource management algorithm is to minimize the upper bound of $\Delta_V\left(\mathbf{\Theta}\left(t\right)\right)$ in the right-hand side of (18) at each

Algorithm 1 The Online Joint Radio and Computational Resource Management Algorithm

```
At the beginning of the $t$th time slot, obtain $\mathbf{\Theta}(t)$, $\{\Gamma_i(t)\}$, and $\{A_i(t)\}$.
Determine $\mathbf{f}(t)$, $\mathbf{p}_{\text{tx}}(t)$, $\boldsymbol{\alpha}(t)$, $\mathbf{D}_s(t)$, and $\mathbf{f}_C(t)$ by solving
    $\mathbf{P}_{\text{PTS}} : \min_{\mathbf{X}(t)} \; -\sum_{i\in\mathcal{N}} Q_i(t) D_{\Sigma,i}(t) - \sum_{i\in\mathcal{N}} T_i(t)\left(D_{s,i}(t) - D_{r,i}(t)\right) + V \cdot P_{\Sigma}(t)$
    $\text{s.t.} \quad \boldsymbol{\alpha}(t) \in \tilde{\mathcal{A}} \text{ and } (11) - (13),$
    where $P_{\Sigma}(t) \triangleq \sum_{i\in\mathcal{N}} w_i \left(p_{\text{tx},i}(t) + p_{l,i}(t)\right) + w_{N+1} p_{\text{ser}}(t)$.
Update $\{Q_i(t)\}$ and $\{T_i(t)\}$ according to (1) and (2), respectively.
Set $t = t + 1$.
```

time slot. By doing so, the amount of tasks waiting in the task buffers can be maintained at a low level, meanwhile, the weighted sum power consumption of the mobile devices and the MEC server can be minimized. The proposed algorithm is summarized in Algorithm 1, where a deterministic optimization problem $\mathbf{P}_{\text{PTS}}$ needs to be solved at each time slot. It is worthy to note that the objective function of $\mathbf{P}_{\text{PTS}}$ corresponds to the right-hand side of (18)[3], and all the constraints in $\mathbf{P_2}$ except the task buffer stability constraint in (14) are retained in $\mathbf{P}_{\text{PTS}}$. The optimal solution for $\mathbf{P}_{\text{PTS}}$ will be developed in the next subsection.

B. Optimal Solution For $\mathbf{P}_{\text{PTS}}$

In this subsection, we will develop the optimal solution for $\mathbf{P}_{\text{PTS}}$, including the optimal CPU-cycle frequencies for the local CPUs, the transmit power and bandwidth allocation for computation offloading, as well as the MEC server scheduling and the CPU-cycle frequencies for the CPU cores at the MEC server. It can be seen that $\mathbf{P}_{\text{PTS}}$ can be solved optimally by solving three sub-problems.

Optimal CPU-Cycle Frequencies Of The Local CPUs: It is straightforward to show that the optimal CPU-cycle frequencies for the local CPUs in time slot t can be obtained by solving the following sub-problem $\mathbf{SP_1}$:

$$\begin{aligned} \mathbf{SP_1} : \min_{\mathbf{f}(t)} \; & \sum_{i\in\mathcal{N}} \left(-Q_i(t)\,\tau f_i(t)\, L_i^{-1} + V \cdot w_i \kappa_{\text{mob},i} f_i^3(t)\right) \\ \text{s.t.} \quad & 0 \le f_i(t) \le f_{i,\max}, i \in \mathcal{N}. \end{aligned} \tag{19}$$

[3] The terms that are not affected by $\mathbf{X}(t)$ is omitted in the objective function of $\mathbf{P}_{\text{PTS}}$.

First, since the objective function of $\mathbf{SP_1}$ is convex and its constraints are linear, $\mathbf{SP_1}$ is a convex optimization problem. Besides, as both the objective function and constraints of $\mathbf{SP_1}$ can be decomposed for individual $f_i\left(t\right)$, the optimization of $f_i\left(t\right)$ can be done separately at each mobile device. Therefore, the optimal $f_i^{\star}\left(t\right)$ is achieved at either the stationary point of $-Q_i\left(t\right)\tau f_i\left(t\right)L_i^{-1}+V\cdot w_i\kappa_{\text{mob},i}f_i^3\left(t\right)$ or one of the boundary points, which is given by

$$f_i^{\star}\left(t\right)=\begin{cases}\min\left\{f_{i,\max},\sqrt{\frac{Q_i(t)\tau}{3\kappa_{\text{mob},i}w_iVL_i}}\right\}, & w_i>0\\ f_{i,\max}, & w_i=0\end{cases},i\in\mathcal{N}. \tag{20}$$

***Remark** 2:* Note that $f_i^{\star}\left(t\right)$ is non-decreasing with $Q_i\left(t\right)$, as it is desirable to execute more tasks in order to keep the queue length of the local task buffer small. Besides, $f_i^{\star}\left(t\right)$ decreases with V, L_i, $\kappa_{\text{mob},i}$ and w_i. In particular, with a larger value of V, the weight of $P_{\Sigma}\left(t\right)$ in $\mathbf{P}_{\text{PTS}}$ becomes larger, and thus the local CPU slows down its frequency to reduce power consumption. With a larger value of w_i, the system places more emphasis on minimizing the power consumption at the ith mobile device, which also contributes to the reduction of $f_i^{\star}\left(t\right)$. On the other hand, with a larger value of L_i ($\kappa_{\text{mob},i}$), local execution becomes less effective as more CPU cycles (Joule of energy) will be needed to process per bit of task input, which again leads to a smaller CPU-cycle frequency. It is worthwhile to mention that although $\mathbf{f}^{\star}\left(t\right)$ depends only on $\mathbf{Q}\left(t\right)$, the optimal local CPU speeds are temporally correlated with the optimal computation offloading decisions as will be derived in the sequel, since both of them affect the task buffer dynamics.

Optimal Transmit Power And Bandwidth Allocation: The optimal $\mathbf{p}_{\text{tx}}^{\star}\left(t\right)$ and $\boldsymbol{\alpha}^{\star}\left(t\right)$ can be obtained by solving the following sub-problem:

$$\begin{aligned}\mathbf{SP_2}:\min_{\boldsymbol{\alpha}(t),\mathbf{p}_{\text{tx}}(t)}&-\sum_{i\in\mathcal{N}}\left(Q_i\left(t\right)-T_i\left(t\right)\right)D_{r,i}\left(t\right)+V\cdot\sum_{i\in\mathcal{N}}w_ip_{\text{tx},i}\left(t\right)\\ \text{s.t.}\quad & 0\leq p_{\text{tx},i}\left(t\right)\leq p_{i,\max},i\in\mathcal{N}\text{ and }\boldsymbol{\alpha}\left(t\right)\in\tilde{\mathcal{A}},\end{aligned} \tag{21}$$

which is non-convex in general. In order to solve $\mathbf{SP_2}$, we identify an important property of its optimal solution, as shown in the following lemma.

***Lemma** 2:* For the set of mobile devices with $Q_i\left(t\right)\leq T_i\left(t\right)$ (denoted as $\tilde{\mathcal{N}}\left(t\right)\triangleq\{i|i\in\mathcal{N},Q_i\left(t\right)\leq T_i\left(t\right)\}$), the optimal transmit power and bandwidth allocation are given by $p_{\text{tx},i}^{\star}\left(t\right)=0$ and $\alpha_i^{\star}\left(t\right)=\epsilon_A$, $i\in\tilde{\mathcal{N}}\left(t\right)$.

Proof: For arbitrary $i\in\tilde{\mathcal{N}}\left(t\right)$, as $D_{r,i}\left(t\right)$ is a non-decreasing function of $p_{\text{tx},i}\left(t\right)$, the term $-\left(Q_i\left(t\right)-T_i\left(t\right)\right)D_{r,i}\left(t\right)+V\cdot w_ip_{\text{tx},i}\left(t\right)$ in the objective function of $\mathbf{SP_2}$ is non-decreasing with $p_{\text{tx},i}\left(t\right)$, i.e., $p_{\text{tx},i}^{\star}\left(t\right)=0$. Besides, with $p_{\text{tx},i}^{\star}\left(t\right)=0$, $\alpha_i\left(t\right)$ has no contribution to the value of the objective function in $\mathbf{SP_2}$. Thus, the portion of bandwidth exceeding ϵ_A that is originally allocated for the ith mobile device can be re-allocated for the devices in $\tilde{\mathcal{N}}^{\rm c}(t) \triangleq \mathcal{N} \setminus \tilde{\mathcal{N}}(t)$, i.e., $\alpha_i^\star(t) = \epsilon_A, i \in \tilde{\mathcal{N}}(t)$. ■

Lemma 2 implies that a mobile device will offload only when the amount of tasks in the local task buffer is greater than that in its task buffer at the server. Intuitively, this is because when $T_i(t) \geq Q_i(t)$, the amount of tasks waiting in the MEC server is relatively large, and offloading tasks not only incurs transmit power consumption, but also brings detrimental effect on the execution delay, and thus it is inferior to local execution. Hence, by ruling out the mobile devices in $\tilde{\mathcal{N}}(t)$, we simplify $\mathbf{SP_2}$ as follows:

$$\begin{aligned} \mathbf{SP_2'}\text{:} \min_{\alpha_i(t), p_{{\rm tx},i}(t), i \in \tilde{\mathcal{N}}^{\rm c}(t)} & - \sum_{i \in \tilde{\mathcal{N}}^{\rm c}(t)} \left(Q_i(t) - T_i(t)\right) D_{r,i}(t) + V \cdot \sum_{i \in \tilde{\mathcal{N}}^{\rm c}(t)} w_i p_{{\rm tx},i}(t) \\ \text{s.t.} \quad & 0 \leq p_{{\rm tx},i}(t) \leq p_{i,\max}, i \in \tilde{\mathcal{N}}^{\rm c}(t) \\ & \alpha_i(t) \geq \epsilon_A, i \in \tilde{\mathcal{N}}^{\rm c}(t), \sum_{i \in \tilde{\mathcal{N}}^{\rm c}(t)} \alpha_i(t) \leq 1 - |\tilde{\mathcal{N}}(t)| \cdot \epsilon_A. \end{aligned} \tag{22}$$

Since $D_i(t) \triangleq \tau\omega \log_2\left(1 + \Gamma_i(t) p_{{\rm tx},i}(t) (N_0\omega)^{-1}\right)$ is concave with respect to $p_{{\rm tx},i}(t)$ and $D_{r,i}(t)$ is a perspective function of $D_i(t)$, i.e., $D_{r,i}(t) = \alpha_i(t) D_i(t)|_{p_{{\rm tx},i}(t)/\alpha_i(t)}$, $D_{r,i}(t)$ is jointly concave with respect to $\alpha_i(t)$ and $p_{{\rm tx},i}(t)$. Therefore, $\mathbf{SP_2'}$ is a convex optimization problem, and thus standard convex algorithms, such as the *interior point method* [38], can be applied for the optimal solution. However, generic convex algorithms suffer from relatively high complexity, as they are developed for general convex problems and do not make full use of the problem structures. Motivated by this, we propose to solve $\mathbf{SP_2'}$ by optimizing the transmit power and the bandwidth allocation in an alternating manner, where in each iteration, the optimal transmit powers are obtained in closed forms and the optimal bandwidth allocation is determined by the *Lagrangian method*. Since $\mathbf{SP_2'}$ is convex and its feasible region is a Cartesian product of those of $\{p_{{\rm tx},i}(t)\}$ and $\{\alpha_i(t)\}$, the alternating minimization procedure is guaranteed to converge to the global optimal solution, which is termed as the *Gauss-Seidel method* in literature [39]. Next, we derive the solution in each iteration of the Gauss-Seidel method for fixed bandwidth and transmit power allocation, respectively.

1) Optimal Transmit Power: For a fixed bandwidth allocation, $\{\alpha_i(t)\}, i \in \tilde{\mathcal{N}}^{\rm c}(t)$, the optimal transmit power for mobile devices in $\tilde{\mathcal{N}}^{\mathrm{c}}(t)$ can be obtained by solving

$$\begin{aligned} \mathbf{P}_{\mathrm{PWR}}: \min_{p_{\mathrm{tx},i}(t), i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} & -\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \left(Q_i(t)-T_i(t)\right) D_{r,i}(t) + V\cdot \sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} w_i p_{\mathrm{tx},i}(t) \\ \text{s.t.} \quad & 0\le p_{\mathrm{tx},i}(t)\le p_{i,\max}, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t). \end{aligned} \tag{23}$$

Similar to $\mathbf{SP_1}$, $\mathbf{P}_{\mathrm{PWR}}$ can be decomposed for individual mobile device, and the optimal $p^{\star}_{\mathrm{tx},i}(t)$ is achieved at either the stationary point of $-\left(Q_i(t)-T_i(t)\right)D_{r,i}(t)+V\cdot w_i p_{\mathrm{tx},i}(t)$ or one of the boundary points, which is given in closed form by

$$p^{\star}_{\mathrm{tx},i}(t)=\begin{cases} \min\left\{\alpha_i(t)\,\omega\max\left\{\frac{(Q_i(t)-T_i(t))\tau}{\ln 2\cdot V\cdot w_i}-\frac{N_0}{\Gamma_i(t)},0\right\},p_{i,\max}\right\}, & w_i>0 \\ p_{i,\max}, & w_i=0 \end{cases}, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t). \tag{24}$$

2) Optimal Bandwidth Allocation: For a fixed transmit power allocation $\{p_{\mathrm{tx},i}(t)\}$, $i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)$, the optimal bandwidth allocation can be obtained by solving the following problem:

$$\begin{aligned} \mathbf{P}_{\mathrm{BW}}: \min_{\alpha_i(t), i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \quad & -\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \left(Q_i(t)-T_i(t)\right) D_{r,i}(t) \\ \text{s.t.} \quad & \alpha_i(t)\ge\epsilon_A, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t), \sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}\alpha_i(t)\le 1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A, \end{aligned} \tag{25}$$

which is more challenging as the bandwidth allocation is coupled among different mobile devices. Fortunately, the Lagrangian method offers an effective solution for $\mathbf{P}_{\mathrm{BW}}$. Specifically, the partial Lagrangian of $\mathbf{P}_{\mathrm{BW}}$ can be written as[4]

$$\mathcal{L}\left(\boldsymbol{\alpha}(t),\lambda(t)\right)=-\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}\left(Q_i(t)-T_i(t)\right)D_{r,i}(t)+\lambda(t)\left[\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}\alpha_i(t)-\left(1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A\right)\right], \tag{26}$$

where $\lambda(t)\ge 0$ is the Lagrangian multiplier associated with $\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}\alpha_i(t)\le 1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A$. When $\exists i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)$ such that $p_{\mathrm{tx},i}(t)\Gamma_i(t)>0$, based on the Karush-Kuhn-Tucker (KKT) conditions, the optimal bandwidth allocation $\{\alpha_i^{\star}(t)\}, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)$ and the optimal Lagrangian multiplier $\lambda^{\star}(t)$ should satisfy the following equation set:

$$\begin{cases} \alpha_i^{\star}(t)=\max\{\epsilon_A,\mathcal{R}_i(\lambda^{\star}(t))\}, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t),\lambda^{\star}(t)>0 \\ \sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}\alpha_i^{\star}(t)=1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A. \end{cases} \tag{27}$$

In (27), if $p_{\mathrm{tx},i}(t)\Gamma_i(t)=0$, we define $\mathcal{R}_i(\lambda(t))\triangleq\epsilon_A$; Otherwise, $\mathcal{R}_i(\lambda(t))$ denotes the root of $\frac{\partial\mathcal{L}(\boldsymbol{\alpha}(t),\lambda(t))}{\partial\alpha_i(t)}=-\left(Q_i(t)-T_i(t)\right)\frac{dD_{r,i}(t)}{d\alpha_i(t)}+\lambda(t)=0$ for $\lambda(t)>0$, which is positive and unique

[4] We slightly abuse notation by using $\boldsymbol{\alpha}(t)$ to denote $\alpha_i(t), i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)$.

as $\frac{dD_{r,i}(t)}{d\alpha_i(t)}$ decreases with $\alpha_i(t)$, $\lim_{\alpha_i(t)\to 0^+}\frac{dD_{r,i}(t)}{d\alpha_i(t)} = +\infty$, and $\lim_{\alpha_i(t)\to +\infty}\frac{dD_{r,i}(t)}{d\alpha_i(t)} = 0$. Thus, it suggests a bisection search over $[\lambda_L(t), \lambda_U(t)]$ for the optimal $\lambda^\star(t)$. Here, $\lambda_L(t)$ and $\lambda_U(t)$ can be chosen as

$$\begin{cases} \lambda_L(t) = \max_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \left(Q_i(t) - T_i(t)\right) \frac{dD_{r,i}(t)}{d\alpha_i(t)}|_{\alpha_i(t)=1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A} \\ \lambda_U(t) = \max_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \left(Q_i(t) - T_i(t)\right) \frac{dD_{r,i}(t)}{d\alpha_i(t)}|_{\alpha_i(t)=\epsilon_A}, \end{cases} \tag{28}$$

which satisfy $\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \max\{\epsilon_A, \mathcal{R}_i(\lambda_L(t))\} > 1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A$ and $\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \max\{\epsilon_A, \mathcal{R}_i(\lambda_U(t))\} < 1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A$, respectively. Hence, $\mathcal{R}_i(\lambda(t))$ can be obtained by a bisection search over $\left(0, 1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A\right]$ when $p_{\mathrm{tx},i}(t)\Gamma_i(t) > 0$, and the searching process for the optimal $\lambda^\star(t)$ will be terminated when $|\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \max\{\epsilon_A, \mathcal{R}_i(\lambda(t))\} - (1-|\tilde{\mathcal{N}}(t)|\cdot\epsilon_A)| < \xi$, where ξ is the accuracy of the algorithm. When $p_{\mathrm{tx},i}(t)\Gamma_i(t) = 0, \forall i \in \tilde{\mathcal{N}}^{\mathrm{c}}(t)$, $\alpha_i^\star(t) = \epsilon_A, i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)$ is the optimal bandwidth allocation. Details of the Lagrangian method for $\mathbf{P}_{\mathrm{BW}}$ are summarized in Algorithm 2.

Remark *3:* The proposed Gauss-Seidel method updates the transmit power and bandwidth allocation alternately in each iteration, which will converge to the optimal solution of $\mathbf{SP_2'}$ with a sublinear convergence rate [40]. In each iteration, the major complexity comes from the Lagrangian method for the optimal bandwidth allocation, which employs bisection search for $\lambda^\star(t)$, and will terminate within $\log_2\left(\frac{\lambda_U(t)-\lambda_L(t)}{\lambda_\xi(t)}\right)$ iterations in time slot t ($\lambda_\xi(t)$ corresponds to the accuracy requirement for $\lambda^\star(t)$ given ξ). Besides, in order to search for the optimal $\lambda^\star(t)$, $\log_2\left(\frac{1}{\varsigma}\right)$ evaluations for $\frac{\partial\mathcal{L}(\boldsymbol{\alpha}(t),\lambda(t))}{\partial\alpha_i(t)}$ are needed in order to determine $\mathcal{R}_i(\lambda)$ for each mobile device, where ς is the accuracy requirement. Therefore, Algorithm 2 will terminate within a finite number of evaluations for functions $\{\frac{\partial\mathcal{L}(\boldsymbol{\alpha}(t),\lambda(t))}{\partial\alpha_i(t)}\}_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)}$, which is given by $|\tilde{\mathcal{N}}^{\mathrm{c}}(t)|\log_2\left(\frac{1}{\varsigma}\right)\log_2\left(\frac{\lambda_U(t)-\lambda_L(t)}{\lambda_\xi(t)}\right)$.

3) Optimal CPU-Cycle Frequencies And Scheduling At The MEC Server: After decoupling $\mathbf{f}(t)$, $\mathbf{p}_{\mathrm{tx}}(t)$ and $\boldsymbol{\alpha}(t)$ from $\mathbf{P}_{\mathrm{PTS}}$, we find that the optimal CPU-cycle frequencies for the M CPU cores at the MEC server $\mathbf{f}_C^\star(t)$ and the optimal scheduling decision $\mathbf{D}_s^\star(t)$ can be obtained by solving the following sub-problem:

$$\begin{aligned} \mathbf{SP_3}: \min_{\mathbf{f}_C(t),\mathbf{D}_s(t)} \quad & -\sum_{i\in\mathcal{N}} T_i(t) D_{s,i}(t) + V\cdot w_{N+1}\sum_{m\in\mathcal{M}} \kappa_{\mathrm{ser},m} f_{C,m}^3(t) \\ \text{s.t.} \quad & 0 \le f_{C,m}(t) \le f_{C_m,\max}, m\in\mathcal{M} \\ & \sum_{n\in\mathcal{N}} D_{s,n}(t) L_n \le \sum_{m\in\mathcal{M}} f_{C,m}(t)\tau, D_{s,i}(t)\ge 0, i\in\mathcal{N}. \end{aligned} \tag{29}$$

It is not difficult to verify that $\mathbf{SP_3}$ is a convex problem, and thus standard convex algorithms

Algorithm 2 Lagrangian Method for $\mathbf{P}_{\mathrm{BW}}$

```
Set $\xi = 10^{-7}$, $\tilde{\lambda}_L = \lambda_L(t)$, $\tilde{\lambda}_U = \lambda_U(t)$, $l = 0$, $I_{\max} = 200$, $\epsilon_A = 10^{-4}$, $\alpha_i(t) = \epsilon_A, i \in \tilde{\mathcal{N}}^{\mathrm{c}}(t)$.

While $|\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \alpha_i(t) - \left(1 - |\tilde{\mathcal{N}}(t)| \cdot \epsilon_A\right)| \geq \xi$ and $l \leq I_{\max}$ do
    $\tilde{\lambda} = \frac{1}{2}\left(\tilde{\lambda}_L + \tilde{\lambda}_U\right)$ and $l = l + 1$.
    Set $\alpha_i(t) = \max\{\epsilon_A, \mathcal{R}_i\left(\tilde{\lambda}\right)\}$, $i \in \tilde{\mathcal{N}}^{\mathrm{c}}(t)$.
    If $\sum_{i\in\tilde{\mathcal{N}}^{\mathrm{c}}(t)} \alpha_i(t) > 1 - |\tilde{\mathcal{N}}(t)| \cdot \epsilon_A$ then
        $\tilde{\lambda}_L = \tilde{\lambda}$.
    Else
        $\tilde{\lambda}_U = \tilde{\lambda}$.
    Endif
Endwhile
```

can be used. Interestingly, by identifying an important structure of the optimal solution as shown in the following lemma, we are able to solve $\mathbf{SP_3}$ in closed form.

***Lemma** 3:* For any feasible $\mathbf{f}_C(t)$, there exists an optimal solution for $\mathbf{SP_3}$ where at most one mobile device is scheduled, i.e., $|\mathbf{D}_s(t)|_0 \leq 1$, and the mobile device being scheduled is the one with the highest value of $T_i(t) L_i^{-1}$,[5] whose device index is denoted as $i_{\mathcal{N}}^{\max}$.

Proof: Please refer to Appendix B. ■

Based on Lemma 3, we have $D_{s,i}^{\star}(t) = 0, i \neq i_{\mathcal{N}}^{\max}$. Therefore, we can simplify $\mathbf{SP_3}$ as

$$\begin{aligned} \mathbf{SP_3'}: \min_{\mathbf{f}_C(t), D_{s,i_{\mathcal{N}}^{\max}}(t)} \quad & -T_{i_{\mathcal{N}}^{\max}}(t) D_{s,i_{\mathcal{N}}^{\max}}(t) + V \cdot w_{N+1} \sum_{m\in\mathcal{M}} \kappa_{\mathrm{ser},m} f_{C,m}^3(t) \\ \mathrm{s.t.} \quad & 0 \leq f_{C,m}(t) \leq f_{C_m,\max}, m \in \mathcal{M} \\ & D_{s,i_{\mathcal{N}}^{\max}}(t) L_{i_{\mathcal{N}}^{\max}} \leq \sum_{m\in\mathcal{M}} f_{C,m}(t)\tau, D_{s,i_{\mathcal{N}}^{\max}}(t) \geq 0, \end{aligned} \tag{30}$$

and its optimal solution is given by the following corollary.

[5] $|\mathbf{x}|_0$ denotes the number of non-zero elements in the vector $\mathbf{x}$. When there are multiple devices with the same maximum value of $T_i(t) L_i^{-1}$, we schedule the one with the smallest device index, which preserves optimality. Therefore, we assume the mobile device with the highest value of $T_i(t) L_i^{-1}$ is unique in the following for ease of presentation.

Corollary *1:* The optimal solution for $\mathbf{SP}_{\mathbf{3}}'$ (also for $\mathbf{SP}_{\mathbf{3}}$) is given as

$$f_{C,m}^{\star}\left(t\right)=\begin{cases}\min\left\{f_{C_m,\max},\sqrt{\frac{T_{i_{\mathcal{N}}^{\max}}\left(t\right)\tau}{3Vw_{N+1}L_{i_{\mathcal{N}}^{\max}}\kappa_{\text{ser},m}}}\right\}, & w_{N+1}>0\\ f_{C_m,\max}, & w_{N+1}=0\end{cases},m\in\mathcal{M}. \tag{31}$$

and $D_{s,i_{\mathcal{N}}^{\max}}^{\star}\left(t\right)=L_{i_{\mathcal{N}}^{\max}}^{-1}\sum_{m\in\mathcal{M}}f_{C,m}^{\star}\left(t\right)\tau$ and $D_{s,i}^{\star}\left(t\right)=0,i\neq i_{\mathcal{N}}^{\max}$.

Proof: Since $T_{i_{\mathcal{N}}^{\max}}\left(t\right)\geq 0$, there exists an optimal solution for $\mathbf{SP}_{\mathbf{3}}'$ such that $D_{s,i_{\mathcal{N}}^{\max}}\left(t\right)L_{i_{\mathcal{N}}^{\max}}=\sum_{m\in\mathcal{M}}f_{C,m}\left(t\right)\tau$. Thus, by substituting $D_{s,i_{\mathcal{N}}^{\max}}\left(t\right)=L_{i_{\mathcal{N}}^{\max}}^{-1}\sum_{m\in\mathcal{M}}f_{C,m}\left(t\right)\tau$ into the objective function, we find that $f_{C,m}^{\star}\left(t\right)$ is the optimal solution for

$$\min_{0\leq f_{C,m}(t)\leq f_{C_m,\max}}-T_{i_{\mathcal{N}}^{\max}}\left(t\right)L_{i_{\mathcal{N}}^{\max}}^{-1}f_{C,m}\left(t\right)\tau+V\cdot w_{N+1}\kappa_{\text{ser},m}f_{C,m}^{3}\left(t\right), \tag{32}$$

which is given by (31). Together with Lemma 3, we complete the proof. ■

Remark *4:* One main benefit of the proposed online algorithm is that it does not require prior information on the computation task arrival and wireless channel fading processes, which makes it also applicable for unpredictable environments. Besides, the proposed algorithm is of low complexity, as at each time slot, the optimal CPU-cycle frequencies of the local and server CPUs, as well as the MEC server scheduling decision, are obtained in closed forms, while the computation offloading policy is determined by an efficient alternating minimization algorithm. Furthermore, as will be shown in the next section, the achievable performance of the proposed algorithm can be analytically characterized, which helps to demonstrate its asymptotic optimality.

C. A Delay-Improved Mechanism

In Algorithm 1, the system operation is determined by the optimal solution of the per-time slot problem $\mathbf{P}_{\text{PTS}}$, which excludes the task buffer underflow constraint, i.e., some of the departure functions, either $D_{\Sigma,i}\left(t\right)$ or $D_{s,i}\left(t\right)$, may be greater than the actual amount of tasks in the corresponding buffers. In particular, as there is only one mobile device being scheduled by the MEC server at each time slot according to Lemma 3 and Corollary 1, part of the available CPU cycles at the MEC server may be wasted and the excessive computational resource can be re-allocated for other devices. Inspired by this observation, we propose a delay-improved mechanism for Algorithm 1 in this subsection.

In the proposed delay-improved mechanism, the decision center maintains a set of virtual task buffers $\mathbf{\Theta}_{vir}\left(t\right)$ with $\mathbf{\Theta}_{vir}\left(0\right)=\mathbf{0}$, and tracks the states of the actual task buffers $\mathbf{\Theta}_{act}\left(t\right)$. The knowledge of $\mathbf{\Theta}_{vir}\left(t\right)$ is used to compute the optimal solution of the per-time slot problem $\mathbf{X}^{\star}\left(t\right)$

according to Algorithm 1. We modify the MEC server scheduling decision $\mathbf{D}_s^{\star}(t)$ in $\mathbf{X}^{\star}(t)$ as $\tilde{\mathbf{D}}_s(t)$ in the actual implementation according to the procedures shown in Algorithm 3.

Algorithm 3 The Delay-Improved Mechanism

```
1: If $\sum_{m\in\mathcal{M}} f_{C,m}^{\star}(t)\tau \leq T_{i_{\mathcal{N}}^{\max},act}(t) L_{i_{\mathcal{N}}^{\max}}$, set $\tilde{\mathbf{D}}_s(t) = \mathbf{D}_s^{\star}(t)$ and exit, where $i_{\mathcal{N}}^{\max} = \arg\max_{i\in\mathcal{N}} T_{i,vir}(t) L_i^{-1}$; Otherwise, go to Line 2.
2: Sort the mobile devices according to the descending order of $T_{i,vir}(t) L_i^{-1}$ and let $[i]$ denote the mobile device with the $i$th highest value of $T_{i,vir}(t) L_i^{-1}$, i.e., $[1] = i_{\mathcal{N}}^{\max}$. (If there are multiple devices with the same values of $T_{i,vir}(t) L_i^{-1}$, sort them according to the ascending order of their device indices.)
3: If $\sum_{i\in\mathcal{N}} T_{i,act}(t) L_i > \sum_{m\in\mathcal{M}} f_{C,m}^{\star}(t)\tau$, determine the number of scheduled mobile devices as $n_{\text{ser}}(t) = \min\left\{n \middle| \sum_{i=1}^{n} T_{[i],act}(t) L_{[i]} > \sum_{m\in\mathcal{M}} f_{C,m}^{\star}(t)\tau\right\}$; Otherwise, $n_{\text{ser}}(t) = N$.
4: Re-allocate the CPU cycles offered by the MEC server according to
   $$\tilde{D}_{s,[n]}(t) = \begin{cases} T_{[n],act}(t), & n < n_{\text{ser}}(t) \\ \left(\sum_{m\in\mathcal{M}} f_{C,m}^{\star}(t)\tau - \sum_{i<n_{\text{ser}}(t)} T_{[i],act}(t) L_{[i]}\right) L_{[n]}^{-1}, & n = n_{\text{ser}}(t) \\ 0, & \text{otherwise.} \end{cases}$$
```

In the following proposition, we show the performance of the delay-improved mechanism, including the average weighted sum power consumption and the average execution delay, is no worse than that achieved by Algorithm 1.

***Proposition** 1:* The power consumption of the mobile devices and the MEC server under the delay-improved mechanism, i.e., Algorithm 3, is the same as that under Algorithm 1, while the average execution delay, i.e., the average sum queue length, is not increased.

Proof: The property of the power consumption performance is straightforward, while the proof for the delay performance can be obtained by verifying $Q_{i,act}(t) = Q_{i,vir}(t)$ and $T_{i,act}(t) \leq T_{i,vir}(t), i \in \mathcal{N}, t \in \mathcal{T}$ via mathematical induction, which is omitted for brevity. ■

It is worthwhile to emphasize that the proposed delay-improved mechanism is able to secure the benefits of Algorithm 1 as mentioned in Remark 4, while enhancing the average execution delay performance without incurring extra power consumption.

V. Performance Analysis

In this section, we will provide the main theoretical results of this paper, which characterize the upper bounds for the average weighted sum power consumption of the mobile devices and the MEC server, as well as the average sum queue length of the task buffers at the mobile and server sides. Besides, the tradeoff between the average weighted sum power consumption and average execution delay will also be revealed.

To facilitate the analysis, we first define an auxiliary problem $\mathbf{P_3}$ with the same objective function and constraints as $\mathbf{P_2}$, but the task buffer dynamics at the MEC server is modified as

$$T_i\left(t+1\right) = \max\{T_i\left(t\right) - D_{s,i}\left(t\right), 0\} + D_{r,i}\left(t\right), i \in \mathcal{N}. \tag{33}$$

In the following lemma, we demonstrate the relationship between $\mathbf{P_2}$ and $\mathbf{P_3}$ in terms of their feasibility and optimal values of the objective functions.

***Lemma** 4:* $\mathbf{P_2}$ is feasible if and only if $\mathbf{P_3}$ is feasible, and the optimal values of $\mathbf{P_2}$ and $\mathbf{P_3}$, denoted as $P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}}$ and $P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}}$, respectively, are equal, i.e., $P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}} = P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}}$.

Proof: We first show that if $\mathbf{P_3}$ is feasible, then $\mathbf{P_2}$ is also feasible.[6] Denote the system operation under the optimal solution for $\mathbf{P_3}$ as $\{\mathbf{X}^{3\star}\left(t\right)\}$, and the queue lengths induced by this solution in $\mathbf{P_3}$ as $\mathbf{\Theta}_{\mathbf{P_3}}^{3\star}\left(t\right) = \left[\mathbf{Q}_{\mathbf{P_3}}^{3\star}\left(t\right), \mathbf{T}_{\mathbf{P_3}}^{3\star}\left(t\right)\right]$. By applying $\{\mathbf{X}^{3\star}\left(t\right)\}$ to $\mathbf{P_2}$, we can easily show that the queue lengths induced by $\{\mathbf{X}^{3\star}\left(t\right)\}$ in $\mathbf{P_2}$, $\mathbf{\Theta}_{\mathbf{P_2}}^{3\star}\left(t\right) = \left[\mathbf{Q}_{\mathbf{P_2}}^{3\star}\left(t\right), \mathbf{T}_{\mathbf{P_2}}^{3\star}\left(t\right)\right]$, satisfies $Q_{\mathbf{P_2},i}^{3\star}\left(t\right) = Q_{\mathbf{P_3},i}^{3\star}\left(t\right)$ and $T_{\mathbf{P_2},i}^{3\star}\left(t\right) \leq T_{\mathbf{P_3},i}^{3\star}\left(t\right), i \in \mathcal{N}, t \in \mathcal{T}$, i.e., $\mathbf{P_2}$ is also feasible. Besides, the average weighted sum power consumption under $\{\mathbf{X}^{3\star}\left(t\right)\}$ in $\mathbf{P_2}$ equals $P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}}$, i.e., $P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}} \leq P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}}$.

Next, suppose $\mathbf{P_2}$ is feasible and its optimal solution is given by $\{\mathbf{X}^{2\star}\left(t\right)\}$, which can be utilized to construct a solution $\{\tilde{\mathbf{X}}\left(t\right)\}$ for $\mathbf{P_3}$ by modifying $\{D_{r,i}^{2\star}\left(t\right)\}$ as

$$\tilde{D}_{r,i}\left(t\right) = \min\{D_{r,i}^{2\star}\left(t\right), \max\{\tilde{Q}_{\mathbf{P_3},i}\left(t\right) - D_{l,i}^{2\star}\left(t\right), 0\}\} \leq D_{r,i}^{2\star}\left(t\right), \tag{34}$$

where $\tilde{\mathbf{\Theta}}_{\mathbf{P_3}}\left(t\right) = \left[\tilde{\mathbf{Q}}_{\mathbf{P_3}}\left(t\right), \tilde{\mathbf{T}}_{\mathbf{P_3}}\left(t\right)\right]$ denotes the queue lengths of the task buffers in $\mathbf{P_3}$ induced by $\{\tilde{\mathbf{X}}\left(t\right)\}$. By mathematical induction, we can show that $\tilde{\mathbf{Q}}_{\mathbf{P_3}}\left(t\right) = \mathbf{Q}_{\mathbf{P_2}}^{2\star}\left(t\right)$ and $\tilde{\mathbf{T}}_{\mathbf{P_3}}\left(t\right) = \mathbf{T}_{\mathbf{P_2}}^{2\star}\left(t\right)$, where $\mathbf{\Theta}_{\mathbf{P_2}}^{2\star}\left(t\right) = \left[\mathbf{Q}_{\mathbf{P_2}}^{2\star}\left(t\right), \mathbf{T}_{\mathbf{P_2}}^{2\star}\left(t\right)\right]$ denotes the queue lengths in $\mathbf{P_2}$ under $\{\mathbf{X}^{2\star}\left(t\right)\}$. Thus, $\mathbf{P_3}$ is also feasible as the transmit powers at the mobile devices for realizing $\tilde{D}_{r,i}\left(t\right)$ are decreased since $\tilde{D}_{r,i}\left(t\right) \leq D_{r,i}^{2\star}\left(t\right)$, i.e., $P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}} \leq \overline{P}_{\Sigma,\mathbf{P_3}}^{\tilde{\mathbf{X}}} \leq P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}}$, where $\overline{P}_{\Sigma,\mathbf{P_3}}^{\tilde{\mathbf{X}}}$ is the value of the objective function in $\mathbf{P_3}$ under $\tilde{\mathbf{X}}\left(t\right)$. As a result, we have $P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}} = P_{\Sigma,\mathbf{P_3}}^{\mathrm{opt}}$. ■

[6] As the instantaneous constraints, i.e., (11)-(13) and $\boldsymbol{\alpha}\left(t\right) \in \tilde{\mathcal{A}}, t \in \mathcal{T}$, are the same for both problems, it is sufficient to verify the mean rate stability of the task buffers in order to show the feasibility of $\mathbf{P_2}$.

In the following lemma, we show there exists a stationary and randomized policy, in which the system operations are i.i.d. among different time slots, that behaves arbitrarily close to the optimal solution for $\mathbf{P_3}$.

***Lemma** 5:* If $\mathbf{P_3}$ is feasible, then for any $\delta > 0$, there exists a stationary and randomized policy Π that satisfies all the instantaneous constraints, and

$$\begin{cases} \mathbb{E}\left[P_{\Sigma}^{\Pi}\left(t\right)\right] \leq P_{\Sigma,\mathbf{P_3}}^{\text{opt}} + \delta \\ \lambda_i \leq \mathbb{E}\left[D_{\Sigma,i}^{\Pi}\left(t\right)\right] + \delta, i \in \mathcal{N} \\ \mathbb{E}\left[D_{r,i}^{\Pi}\left(t\right)\right] \leq \mathbb{E}\left[D_{s,i}^{\Pi}\left(t\right)\right] + \delta, i \in \mathcal{N}. \end{cases} \tag{35}$$

Proof: The proof can be obtained by Theorem 4.5 in [37], which is omitted for brevity. ■

Based on Lemma 4 and Lemma 5, the worst-case average weighted sum power consumption and average sum queue length under Algorithm 1 and Algorithm 3 are derived in the following theorem.

***Theorem** 1:* Assume that $\mathbf{P_2}$ is feasible, then under Algorithm 1 (also Algorithm 3), we have:

1) The average weighted sum power consumption satisfies:

$$\overline{P}_{\Sigma}^{\star} \leq P_{\Sigma,\mathbf{P_2}}^{\text{opt}} + C \cdot V^{-1}, \tag{36}$$

where $\overline{P}_{\Sigma}^{\star}$ is the average weighted sum power consumption under the proposed algorithm.

2) For arbitrary $i \in \mathcal{N}$, $Q_i\left(t\right)$ and $T_i\left(t\right)$ are mean rate stable.

3) Suppose there exist $\epsilon > 0$ and $\Psi\left(\epsilon\right)$ ($P_{\Sigma,\mathbf{P_2}}^{\text{opt}} \leq \Psi\left(\epsilon\right) \leq \sum_{i\in\mathcal{N}} w_i \left(\kappa_{\text{mob},i} f_{i,\max}^3 + p_{i,\max}\right) + w_{N+1} \sum_{m\in\mathcal{M}} \kappa_{\text{ser},m} f_{C_m,\max}^3$) , and a stationary and randomized algorithm $\tilde{\Pi}$ that satisfies $\boldsymbol{\alpha}\left(t\right) \in \tilde{\mathcal{A}}, t \in \mathcal{T}$, (11)-(13), and the following Slater conditions [37]:

$$\begin{cases} \mathbb{E}\left[P_{\Sigma}^{\tilde{\Pi}}\left(t\right)\right] = \Psi\left(\epsilon\right) \\ \lambda_i \leq \mathbb{E}\left[D_{\Sigma,i}^{\tilde{\Pi}}\left(t\right)\right] - \epsilon, i \in \mathcal{N} \\ \mathbb{E}\left[D_{r,i}^{\tilde{\Pi}}\left(t\right)\right] \leq \mathbb{E}\left[D_{s,i}^{\tilde{\Pi}}\left(t\right)\right] - \epsilon, i \in \mathcal{N}, \end{cases} \tag{37}$$

then the average sum queue length of the task buffers of all the mobile devices satisfies:

$$\sum_{i\in\mathcal{N}} \overline{q}_{\Sigma,i} \leq \frac{C + V\left(\Psi\left(\epsilon\right) - P_{\Sigma,\mathbf{P_2}}^{\text{opt}}\right)}{\epsilon}. \tag{38}$$

Proof: Please refer to Appendix C. ■

***Remark** 5:* Theorem 1 shows that under the proposed online joint radio and computational resource management algorithm, the worst-case average weighted sum power consumption of the MEC system decreases inversely proportional to V, which demonstrates the asymptotic optimality of the proposed algorithm. It is also shown that the upper bound of the average sum queue length, i.e., the execution delay according to Little's Law, increases linearly with V, i.e., there exists an $[O(1/V), O(V)]$ tradeoff between these two objectives. Thus, we can balance the system's weighted sum power consumption and execution delay by adjusting V: For delay-sensitive applications, we can use a small value of V; while for energy-sensitive networks and delay-tolerant applications, a large value of V can be adopted.

VI. Simulation Results

In simulations, we assume N mobile devices are located at an equal distance of 150 m from the MEC server. The small-scale fading channel power gains are exponentially distributed with unit mean, i.e., $\gamma_i(t) \sim \mathrm{Exp}(1), i \in \mathcal{N}$. Besides, $\tau = 1$ ms, $\omega = 10$ MHz, $N_0 = -174$ dBm/Hz, $g_0 = -40$ dB, $d_0 = 1$ m, $\theta = 4$, $w_i = 1$, $\kappa_{\mathrm{mob},i} = 10^{-27}$, $f_{i,\max} = 1$ GHz, $p_{i,\max} = 500$ mW, $A_i(t)$ is uniformly distributed within $[0, A_{i,\max}]$, and $L_i = 737.5$ cycles/bit, $i \in \mathcal{N}$ [30]. In addition, we set $f_{C_m,\max} = 2.5$ GHz, $\kappa_{\mathrm{ser},m} = 10^{-27}$, $m \in \mathcal{M}$. The simulation results are averaged over 10000 time slots.

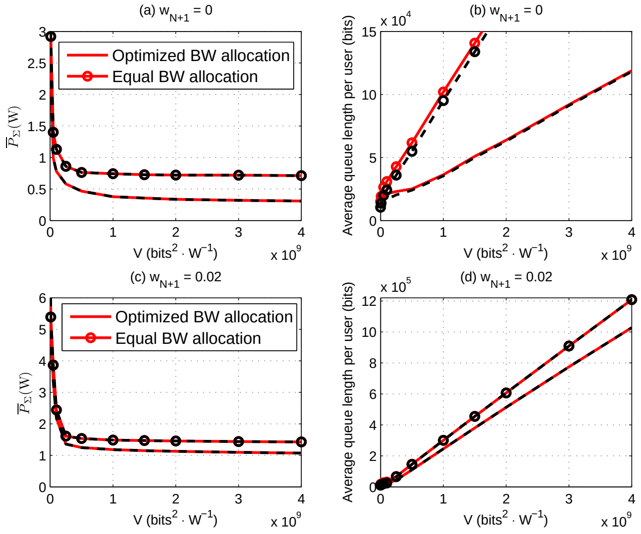

Fig. 2. Average weighted sum power consumption/average sum queue length of the task buffers per mobile device vs. V, $N = 5$, $M = 8$, and $A_{i,\max} = 8$ kbits. (The solid curves and dash curves correspond to Algorithm 1 and Algorithm 3, respectively, while the curves marked with circles are obtained by setting $\alpha_i(t) = 1/N, i \in \mathcal{N}$ in the proposed algorithms.)

We first validate the theoretical results derived in Theorem 1 for the proposed algorithms in Fig. 2. Two scenarios with $w_{N+1} = 0$ and 0.02, respectively, are considered. The average weighted sum power consumption for the scenario with $w_{N+1} = 0$ ($w_{N+1} = 0.02$) is shown in Fig. 2a) (Fig.

2c)), while the average sum queue length of the task buffers per mobile device, i.e., $\sum_{i\in\mathcal{N}}\overline{q}_{\Sigma,i}/N$, for $w_{N+1}=0$ ($w_{N+1}=0.02$) is shown in Fig. 2b) (Fig. 2d)). It can be observed from Fig. 2a) and Fig. 2c) that the average weighted sum power consumption decreases inversely proportional to the control parameter V and converges to $P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}}$ when V is sufficiently large. Meanwhile, as shown in Fig. 2b) and Fig. 2d), the average sum queue length of the task buffers increases linearly with V for both Algorithm 1 and its delay-improved version (Algorithm 3), which verifies the $[O(1/V), O(V)]$ tradeoff between the average weighted sum power consumption and the average execution delay as demonstrated in Theorem 1 and Remark 5. Furthermore, we see that increasing w_{N+1} results in a higher average sum queue length, which comes from the slowdown of the server CPU frequencies.

The proposed algorithm and its delay-improved version with equal bandwidth allocation, i.e., $\alpha_i(t)=1/N, i\in\mathcal{N}$, are also evaluated in Fig. 2. We see the performance of both the average weighted sum power consumption and the average sum queue length of the task buffers are deteriorated compared to those achieved with optimized bandwidth allocation. This demonstrates the importance of a joint consideration on the radio and computational resource management for multi-user MEC systems. Besides, it is shown that the average weighted sum power consumptions under both Algorithm 1 and Algorithm 3 are the same, while the average sum queue length of the task buffers under the delay-improved mechanism is reduced, which agrees with Proposition 1. Such queue length (i.e., execution delay) reduction is more obvious when w_{N+1} and V get closer to zero. This is because under these circumstances, the frequencies of the server CPU cores are much higher than what are needed for serving the mobile device with the highest value of $T_i(t)L_i^{-1}$. In other words, more computational resource provided by the MEC server is wasted under Algorithm 1 and can be re-allocated for other mobile devices. We only focus on Algorithm 3 in the following, as it always achieves improved performance compared to Algorithm 1.

In Fig. 3, we reveal the impacts of the computation capability of the MEC server by setting $w_{N+1}=0$ and varying the number of CPU cores M. In these scenarios, the power consumption of the MEC server is not addressed in the optimization and thus the server CPU cores will operate at their maximum frequencies. As can be seen from Fig. 3, the system performance improves as the value of M increases, i.e., the power consumption of the mobile devices decreases for a given average execution delay.[7] Also, by increasing V from 1×10^6 to 7×10^9 $\mathrm{bit}^2\cdot\mathrm{W}^{-1}$, the power consumption decreases significantly for all curves. Nonetheless, the behaviors of the curves for

[7] The average execution delay is calculated by $\sum_{i\in\mathcal{N}}\overline{q}_{\Sigma,i}/\sum_{i\in\mathcal{N}}\lambda_i$ (time slots) according to Little's Law.

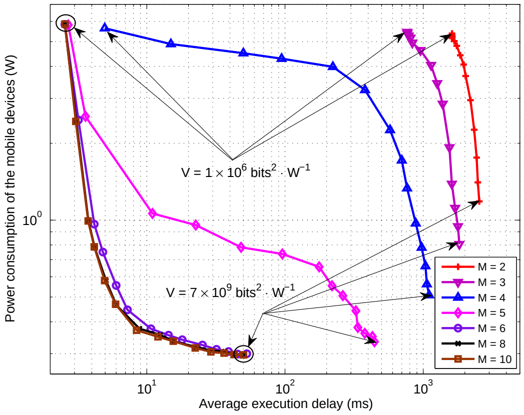

Fig. 3. Power consumption of the mobile devices vs. average execution delay, $w_{N+1} = 0$, $N = 5$, and $A_{i,\max} = 8$ kbits.

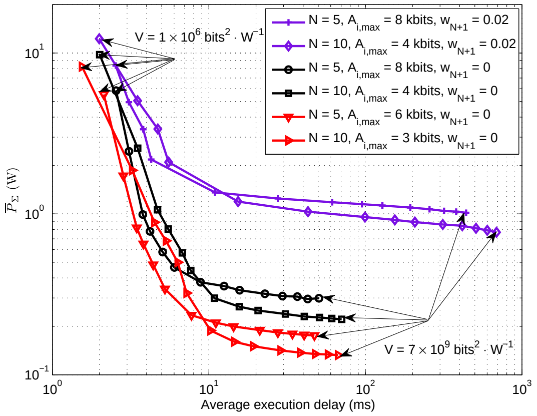

Fig. 4. Weighted sum power consumption of the MEC system vs. average execution delay, $M = 8$.

different values of M are substantially different. For a relatively large value of M, i.e., $M = 6$, 8 and 10, the average execution delay decreases sharply from 52 to 2.6 ms as V decreases, and the power and delay performance follows the $[O(1/V), O(V)]$ tradeoff. However, for a relatively small value of M, the average execution delay has minor changes as V is adjusted, i.e., around 2000 and 1000 ms for $M = 2$ and 3, respectively, and the speed of power reduction increases as the average execution delay increases. This is because with insufficient computational resource at the MEC server, the task buffers cannot be stabilized even with $V = 1 \times 10^6\ \mathrm{bit}^2 \cdot \mathrm{W}^{-1}$ within 10000 time slots. Similar phenomenon exists for the curves with a medium value of M (such as 4 and 5) when a large value of V is used. In addition, the performance gain achieved by increasing M from 6 to 10 is negligible, which indicates that the computation capability of the MEC server should be chosen carefully to balance the system cost and the achievable performance. In particular, once the system is constrained by the radio resource, there is no need to deploy too much computational resource.

By varying $A_{i,\max}$ and N, we show the relationship between the average weighted sum power consumption and the average execution delay in Fig. 4. Similar to Fig. 2 and Fig. 3, the average execution delay increases as the weighted sum power consumption decreases, which indicates that a proper V should be chosen to balance the two desirable objectives. For instance, with $N = 5$, $A_{i,\max} = 8$ kbits and $w_{N+1} = 0$, if the average execution delay requirement is 20 ms, $V = 3 \times 10^9\ \mathrm{bits}^2 \cdot \mathrm{W}^{-1}$ can be chosen, and the sum power consumption of the mobile devices will be about 0.3 W. Besides, with a given execution delay, the average weighted sum power consumption increases with the computation task arrival rate, which fits the intuition,

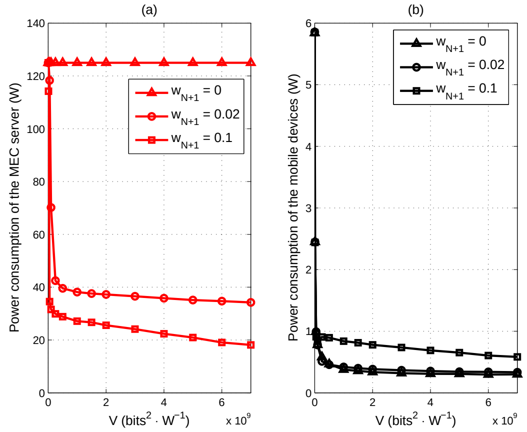

Fig. 5. Power consumption of the MEC server/mobile devices vs. V, $M = 8$, $N = 5$ and $A_{i,\max} = 8$ kbits.

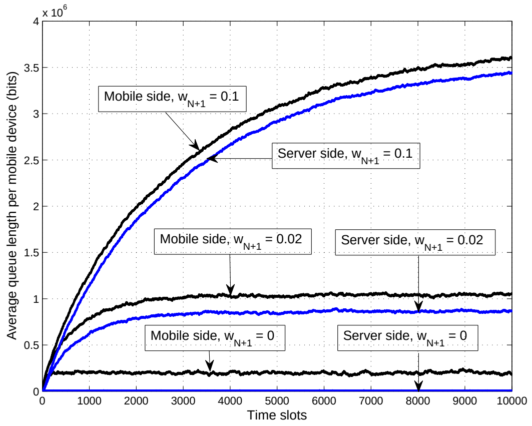

Fig. 6. Average queue length per mobile device vs. time, $N = 5$, $M = 8$, $A_{i,\max} = 8$ kbits, and $V = 7 \times 10^9\ \text{bits}^2 \cdot \text{W}^{-1}$.

i.e., as the workload of the MEC system becomes more intensive, more power is needed for keeping the task buffers stable. In addition, given a total task arrival rate in the MEC system, increasing the number of mobile devices while decreasing the task arrival rate at individual device results in a lower weighted sum power consumption when V is tuned sufficiently large, i.e., the proposed algorithm is approaching the optimal performance. This is due to the increased multi-user diversity gain and the availability of extra local processing units.

We show the impacts of w_{N+1} on the power consumption of the MEC server and the mobile devices in Fig. 5. It can be observed that by increasing w_{N+1}, the power consumption of the MEC server decreases while that of the mobile devices increases for a given value of V, which confirms that the power consumption of the mobile devices and the MEC server can be balanced by adjusting the weighting factors. To be more specific, the power consumption of the MEC server is a constant for $w_{N+1} = 0$ due to the optimal frequencies of the server CPU cores derived in (31). By increasing w_{N+1} from 0 to 0.02, the power consumption of the MEC server is greatly reduced while that of the mobile devices has only minor increase, which leads to higher queue lengths at the task buffers. By increasing w_{N+1} from 0.02 to 0.1, the power consumption of the MEC server further decreases and that of the mobile devices increases noticeably. In this case, the local CPUs needs to increase their CPU-cycle frequencies to compensate the processing speed reduction at the MEC server in order to keep the task buffers stable. For better demonstration, we show the evolution of the average queue lengths at the mobile devices (i.e., $\sum_{i\in\mathcal{N}} Q_i(t)/N$) and the MEC server (i.e., $\sum_{i\in\mathcal{N}} T_i(t)/N$) over time in Fig. 6. We see for $w_{N+1} = 0$, the amount of waiting tasks in the server side buffers is near zero as the MEC server is able to execute all tasks offloaded from the mobile devices, and the queue length of the local task buffers is also maintained at a small level as most tasks can be offloaded to and executed by the MEC server. For $w_{N+1} = 0.02$, the amount of tasks in the buffers increases due to the decrease of CPU frequencies at the MEC server. With $w_{N+1} = 0.1$, the queue lengths of the task buffers at both the mobile devices and the MEC server are much higher than those achieved by $w_{N+1} = 0.02$ and keep increasing as time elapses, which implies that the system cannot converge to its stabilized performance within the time window used in simulations, and in turn lifts up the local CPU-cycle frequencies and the power consumption of the mobile devices.

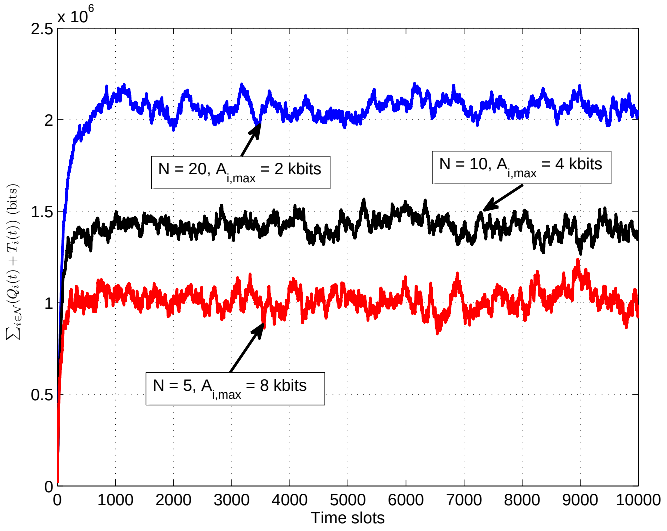

Fig. 7. Sum queue length of the task buffers vs. time slots, $M = 8$, $\sum_{i\in\mathcal{N}} \lambda_i = 20$ kbits, $V = 7 \times 10^9\ \mathrm{bit}^2 \cdot \mathrm{W}^{-1}$.

Finally, we demonstrate the impact of the network size on the convergence time under the proposed algorithm by tracing the sum queue length of the task buffers for systems with different numbers of mobile devices, while keeping the total computation task arrival rate in the MEC system unchanged at $\sum_{i\in\mathcal{N}} \lambda_i = 20$ kbits. It is observed from Fig. 7 that for all the three cases, the sum queue length of the task buffers increases at the beginning, and stabilizes at around 1×10^6 bits, 1.4×10^6 bits, and 2×10^6 bits for $N = 5$, 10, and 20, respectively. Besides, we see that although a larger value of N leads to a longer convergence time, the differences are minor and the task buffers are able to be stabilized within approximately 1000 time slots for all the cases, which indicates that the proposed algorithms are scalable to MEC systems with a reasonable amount of mobile devices.

VII. Conclusions

In this paper, we investigated stochastic joint radio and computational resource management for multi-user mobile-edge computing (MEC) systems. A low-complexity online algorithm based on Lyapunov optimization was proposed, which achieves asymptotic optimality. Furthermore, a delay-improved mechanism was designed to reduce the execution delay. Performance analysis, as well as simulations, explicitly characterize the tradeoff between the average weighted sum power consumption and the average execution delay in a multi-user MEC system. Moreover, the impacts of various parameters were revealed, which confirm the importance of the joint radio and computational resource management, and provide valuable guidelines for the practical deployment of MEC systems. For future investigation, it would be interesting to extend the findings in this work to scenarios with fairness considerations among multiple devices and seek distributed implementation methodologies. Other research problems, such as mobility-aware resource management, dynamic access control and user-server association, should also be investigated.

Appendix

A. Proof for Lemma 1

By squaring both sides of the local task buffer dynamics in (1), we have

$$\begin{aligned} Q_i^2(t+1) &= \left(\max\{Q_i(t) - D_{\Sigma,i}(t), 0\}\right)^2 + A_i^2(t) + 2A_i(t)\cdot\max\{Q_i(t) - D_{\Sigma,i}(t), 0\} \\ &\le \left(Q_i(t) - D_{\Sigma,i}(t)\right)^2 + A_i^2(t) + 2A_i(t)\,Q_i(t) \\ &= Q_i^2(t) - 2Q_i(t)\left(D_{\Sigma,i}(t) - A_i(t)\right) + A_i^2(t) + D_{\Sigma,i}^2(t). \end{aligned} \tag{39}$$

By moving $Q_i^2(t)$ to the left-hand side of (39), dividing both sides by 2, and summing up the inequalities for $i = 1, \cdots, N$, we have

$$\frac{1}{2}\sum_{i\in\mathcal{N}}\left[Q_i^2(t+1) - Q_i^2(t)\right] \le -\sum_{i\in\mathcal{N}} Q_i(t)\left(D_{\Sigma,i}(t) - A_i(t)\right) + \sum_{i\in\mathcal{N}} \frac{A_i^2(t) + D_{\Sigma,i}^2(t)}{2}. \tag{40}$$

Similar manipulations can be made for the MEC server task buffer dynamics in (2) as follows:

$$\begin{aligned} T_i^2(t+1) &\le \left(\max\{T_i(t) - D_{s,i}(t), 0\} + D_{r,i}(t)\right)^2 \\ &= \left(\max\{T_i(t) - D_{s,i}(t), 0\}\right)^2 + D_{r,i}^2(t) + 2D_{r,i}(t)\cdot\max\{T_i(t) - D_{s,i}(t), 0\} \\ &\le \left(T_i(t) - D_{s,i}(t)\right)^2 + D_{r,i}^2(t) + 2D_{r,i}(t)\,T_i(t) \\ &= T_i^2(t) - 2T_i(t)\left(D_{s,i}(t) - D_{r,i}(t)\right) + D_{r,i}^2(t) + D_{s,i}^2(t). \end{aligned} \tag{41}$$

By moving $T_i^2\left(t\right)$ to the left-hand side of (41), dividing both sides by 2, and summing up the inequalities for $i = 1, \cdots, N$, we have

$$\frac{1}{2}\sum_{i\in\mathcal{N}}\left[T_i^2\left(t+1\right) - T_i^2\left(t\right)\right] \leq -\sum_{i\in\mathcal{N}} T_i\left(t\right)\left(D_{s,i}\left(t\right) - D_{r,i}\left(t\right)\right) + \sum_{i\in\mathcal{N}}\frac{D_{r,i}^2\left(t\right) + D_{s,i}^2\left(t\right)}{2}. \tag{42}$$

By summing up (40) and (42), we have

$$\begin{aligned} L\left(\boldsymbol{\Theta}\left(t+1\right)\right) - L\left(\boldsymbol{\Theta}\left(t\right)\right) \leq & -\sum_{i\in\mathcal{N}}\left[Q_i\left(t\right)\left(D_{\Sigma,i}\left(t\right) - A_i\left(t\right)\right) + T_i\left(t\right)\left(D_{s,i}\left(t\right) - D_{r,i}\left(t\right)\right)\right] \\ & + \frac{1}{2}\sum_{i\in\mathcal{N}}\left(A_i^2\left(t\right) + D_{\Sigma,i}^2\left(t\right) + D_{s,i}^2\left(t\right) + D_{r,i}^2\left(t\right)\right). \end{aligned} \tag{43}$$

Since $\log_2\left(1+x\right) \leq \frac{x}{\ln 2}$ and $\log_2^2\left(1+x\right) \leq \frac{2x}{\left(\ln 2\right)^2}$ for $x \geq 0$, we have $\mathbb{E}\left[D_{r,i}^2\left(t\right)|\boldsymbol{\Theta}\left(t\right)\right] \leq \frac{\omega\eta_i}{\ln 2}$ and $\mathbb{E}\left[D_{\Sigma,i}^2\left(t\right)|\boldsymbol{\Theta}\left(t\right)\right] \leq \mathbb{E}\left[\left(\tau f_{i,\max}L_i^{-1} + D_{r,i}\left(t\right)\right)^2|\boldsymbol{\Theta}\left(t\right)\right] \leq \tau^2 f_{i,\max}^2 L_i^{-2} + \eta_i\left(f_{i,\max}L_i^{-1} + \frac{\omega}{\ln 2}\right)$ with $\eta_i = \frac{2g_0\overline{\gamma_i}p_{i,\max}d_0^\theta\tau^2}{\ln 2N_0 d_i^\theta}$. Finally, by adding $V \cdot P_\Sigma\left(t\right)$ at both sides of (43) and taking the expectation on both sides conditioned on $\boldsymbol{\Theta}\left(t\right)$, we can obtain the desired result in (18), where

$$C = \frac{1}{2}\sum_{i\in\mathcal{N}}\left[A_{i,\max}^2 + \left(\sum_{m\in\mathcal{M}} f_{C_m,\max}\tau L_i^{-1}\right)^2 + \left(f_{i,\max}\tau L_i^{-1}\right)^2 + \eta_i\left(f_{i,\max}L_i^{-1} + \frac{2\omega}{\ln 2}\right)\right]. \tag{44}$$

B. Proof for Lemma 3

Since $T_i\left(t\right) \geq 0, i \in \mathcal{N}$, there exists an optimal solution such that $\sum_{n\in\mathcal{N}} D_{s,n}\left(t\right)L_n = \sum_{m\in\mathcal{M}} f_{C,m}\left(t\right)\tau$. For $\sum_{m\in\mathcal{M}} f_{C,m}\left(t\right) = 0$, the optimal scheduling decision $\mathbf{D}_s\left(t\right) = \mathbf{0}$. For $\sum_{m\in\mathcal{M}} f_{C,m}\left(t\right) > 0$, suppose there is an optimal scheduling decision $\mathbf{D}_s\left(t\right)$ with $|\mathbf{D}_s\left(t\right)|_0 \geq 2$. Denote $\mathcal{S} \triangleq \{n|D_{s,n}\left(t\right) > 0, n \in \mathcal{N}\}$ where $|\mathcal{S}| = |\mathbf{D}_s\left(t\right)|_0$. We can construct a new feasible scheduling decision $\hat{\mathbf{D}}_s\left(t\right)$ with $\hat{D}_{s,i_{\mathcal{S}}^{\max}}\left(t\right) = D_{s,i_{\mathcal{S}}^{\max}}\left(t\right) + L_{i_{\mathcal{S}}^{\max}}^{-1}\sum_{j\in\mathcal{S},j\neq i_{\mathcal{S}}^{\max}} D_{s,j}\left(t\right)L_j$ and $\hat{D}_{s,i}\left(t\right) = 0, i \neq i_{\mathcal{S}}^{\max}$, where $i_{\mathcal{S}}^{\max}$ is the mobile device in $\mathcal{S}$ with the highest value of $T_i\left(t\right)L_i^{-1}$. With the constructed solution, the value of the objective function will decrease by

$$-\sum_{i\in\mathcal{N}} T_i\left(t\right)\left(D_{s,i}\left(t\right) - \hat{D}_{s,i}\left(t\right)\right) = \sum_{j\in\mathcal{S},j\neq i_{\mathcal{S}}^{\max}} D_{s,j}\left(t\right)L_{i_{\mathcal{S}}^{\max}}^{-1}\left(T_{i_{\mathcal{S}}^{\max}}\left(t\right)L_j - T_j\left(t\right)L_{i_{\mathcal{S}}^{\max}}\right) \geq 0, \tag{45}$$

i.e., $\hat{\mathbf{D}}_s\left(t\right)$ with $|\hat{\mathbf{D}}_s\left(t\right)|_0 = 1$ performs no worse than $\mathbf{D}_s\left(t\right)$. In addition, if $i_{\mathcal{S}}^{\max} \neq i_{\mathcal{N}}^{\max}$, we can serve the $i_{\mathcal{N}}^{\max}$th device with the CPU cycles that are originally allocated for the $i_{\mathcal{S}}^{\max}$th device, and the value of the objective function will further decrease by $\hat{D}_{s,i_{\mathcal{S}}^{\max}}\left(t\right)L_{i_{\mathcal{N}}^{\max}}^{-1}(T_{i_{\mathcal{N}}^{\max}}\left(t\right)L_{i_{\mathcal{S}}^{\max}} - T_{i_{\mathcal{S}}^{\max}}\left(t\right)L_{i_{\mathcal{N}}^{\max}}) \geq 0$. In other words, there exists an optimal solution for $\mathbf{SP_3}$ for a given $\mathbf{f}_C\left(t\right)$ such that $D_{s,i_{\mathcal{N}}^{\max}}\left(t\right) \geq 0$ and $D_{s,i}\left(t\right) = 0, i \neq i_{\mathcal{N}}^{\max}$.

C. Proof for Theorem 1

Denote the optimal solution for the per-time slot problem in time slot t as $\mathbf{X}^{\star}(t)$. Since the system operation at each time slot is the optimal solution of the per-time slot problem, we have

$$\begin{aligned}\Delta_V(\mathbf{\Theta}(t)) \le\, & C - \mathbb{E}\left[\sum_{i\in\mathcal{N}} Q_i(t)\left(D^{\star}_{\Sigma,i}(t) - A_i(t)\right)|\mathbf{\Theta}(t)\right] \\ & - \mathbb{E}\left[\sum_{i\in\mathcal{N}} T_i(t)\left(D^{\star}_{s,i}(t) - D^{\star}_{r,i}(t)\right)|\mathbf{\Theta}(t)\right] + V\cdot\mathbb{E}\left[P^{\star}_{\Sigma}(t)|\mathbf{\Theta}(t)\right] \qquad (46)\\ & \overset{(a)}{\le} C + \sum_{i\in\mathcal{N}}(Q_i(t)+T_i(t))\,\delta + V\cdot\left(P^{\mathrm{opt}}_{\Sigma,\mathbf{P_3}}+\delta\right),\end{aligned}$$

where (a) holds since the stationary and randomized policy Π is sub-optimal for $\mathbf{P}_{\mathrm{PTS}}$ and also due to the properties of Π as shown in Lemma 5. By taking $\delta\to 0$, we have

$$\Delta_V(\mathbf{\Theta}(t)) \le C + V\cdot P^{\mathrm{opt}}_{\Sigma,\mathbf{P_3}} = C + V\cdot P^{\mathrm{opt}}_{\Sigma,\mathbf{P_2}}. \qquad (47)$$

By taking the expectation on both sides of (47), summing up the inequalities for $t=0,\cdots,T-1$, we have

$$\mathbb{E}\left[L(\mathbf{\Theta}(T))\right] - \mathbb{E}\left[L(\mathbf{\Theta}(0))\right] + V\cdot\sum_{t=0}^{T-1}\mathbb{E}\left[P^{\star}_{\Sigma}(t)\right] \le T\cdot\left(C + V\cdot P^{\mathrm{opt}}_{\Sigma,\mathbf{P_2}}\right), \qquad (48)$$

where $P^{\star}_{\Sigma}(t)$ is the weighted sum power consumption in time slot t under $\mathbf{X}^{\star}(t)$. Since $\mathbf{\Theta}(0)=\mathbf{0}$ and $\mathbb{E}\left[L(\mathbf{\Theta}(T))\right]\ge 0$, by dividing both sides of (48) by T and letting T go to infinite, we have $\overline{P}^{\star}_{\Sigma} \le P^{\mathrm{opt}}_{\Sigma,\mathbf{P_2}} + C\cdot V^{-1}$.

We now proceed to show the mean rate stability of the task buffers. Denote $\Xi(T) \triangleq T\cdot\left(C+V\cdot P^{\mathrm{opt}}_{\Sigma,\mathbf{P_2}}\right) - V\cdot\sum_{t=0}^{T-1}\mathbb{E}\left[P^{\star}_{\Sigma}(t)\right] + \mathbb{E}\left[L(\mathbf{\Theta}(0))\right]$. Based on (48), we have $\mathbb{E}\left[L(\mathbf{\Theta}(T))\right] \le \Xi(T)$, i.e., $\mathbb{E}\left[Q_i^2(T)\right] \le 2\Xi(T)$ and $\mathbb{E}\left[T_i^2(T)\right] \le 2\Xi(T)$. Thus, we have $0 \le \mathbb{E}\left[|Q_i(T)|\right] \le \sqrt{2\Xi(T)}$ and $0 \le \mathbb{E}\left[|T_i(T)|\right] \le \sqrt{2\Xi(T)}, i\in\mathcal{N}$. Since $\lim_{T\to+\infty}\frac{\sqrt{2\Xi(T)}}{T}=0$, we have

$$\lim_{T\to+\infty}\frac{\mathbb{E}\left[|Q_i(T)|\right]}{T} = \lim_{T\to+\infty}\frac{\mathbb{E}\left[|T_i(T)|\right]}{T} = 0, i\in\mathcal{N}, \qquad (49)$$

i.e., $\forall i\in\mathcal{N}$, $Q_i(t)$ and $T_i(t)$ are mean rate stable under Algorithm 1.

In order to show the upper bound of the average sum queue length, we plug the stationary and randomized policy $\tilde{\Pi}$ into the right-hand side of (18), i.e.,

$$\Delta_V(\mathbf{\Theta}(t)) \le C - \epsilon\cdot\sum_{i\in\mathcal{N}}(Q_i(t)+T_i(t)) + V\cdot\Psi(\epsilon). \qquad (50)$$

By taking the expectation on both sides of the inequality, summing up the inequalities for $t = 0, \cdots, T-1$, dividing both sides by $T\epsilon$, and letting T go to infinite, we have

$$\sum_{i\in\mathcal{N}} \overline{q}_{\Sigma,i} \leq \frac{C + V \cdot \left(\Psi\left(\epsilon\right) - \lim\limits_{T\rightarrow+\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[P_{\Sigma}^{\star}\left(t\right)\right]\right)}{\epsilon} \leq \frac{C + V \cdot \left(\Psi\left(\epsilon\right) - P_{\Sigma,\mathbf{P_2}}^{\mathrm{opt}}\right)}{\epsilon}. \qquad (51)$$

With the aid of Proposition 1, we can show Theorem 1 also holds for Algorithm 3.

References

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

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswmi, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *ELSEVIER Future Gener. Comp. Syst.*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[3] European Telecommunications Standards Institute (ETSI), “Mobile-edge computing-Introductory technical white paper,” Sep. 2014.

[4] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks,” *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45-55, Nov. 2014.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “Mobile edge computing: Survey and research outlook,” submitted to *IEEE Commun. Surveys Tuts.*, Jan. 2017. [Online]. Available: https://arxiv.org/pdf/1701.01090v1.pdf

[6] 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.

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

[8] 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.

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

[10] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: Making smartphones last longer with code offload,” in *Proc. ACM Int. Conf. Mobile Syst., Appl., Serv. (MobiSys)*, San Francisco, CA, Jun. 2010, pp. 49-62.

[11] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, Mar. 2012, pp. 945-953.

[12] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569-4581, Sep. 2013.

[13] O. Munoz, A. Iserte, and J. Vidal, “Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738-4755, Oct. 2015.

[14] K. Wang, K. Yang, and C. Magurawalage, “Joint energy minimization and resource allocation in C-RAN with mobile cloud,” *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1-11, Jan. 2016.

[15] Y. Yu, J. Zhang, and K. B. Letaief, “Joint subcarrier and CPU time allocation for mobile edge computing,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, Dec. 2016, pp. 1-6.

[16] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*. vol. PP, no. 99, pp. 1-12, Dec. 2016.

[17] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974-983, Apr. 2015.

[18] 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.

[19] 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.

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

[21] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887-3901, Dec. 2016.

[22] 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.

[23] 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)*, Barcelona, Spain, Jul. 2016, pp. 1451-1455.

[24] Z. Jiang and S. Mao, "Energy delay trade-off in cloud offloading for multi-core mobile devices," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, Dec. 2015, pp. 1-6.

[25] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510-2523, Dec. 2015.

[26] Y. Kim, J. Kwak, and S. Chong, "Dual-side dynamic controls for cost minimization in mobile cloud computing systems," in *Proc. Int. Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, Mumbai, India, May 2015, pp. 443-450.

[27] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. the First Edition of the MCC Workshop on Mobile Cloud Computing*, Helsinki, Finland, Aug. 2012, pp. 13-16.

[28] C. Zhu, V. C. M. Leung, L. T. Yang, and L. Shu, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," *IEEE Trans. Compt.*, vol. 64, no. 7, pp. 1844-1858, Jul. 2015.

[29] L. Chen, O. Villa, and G. Gao, "Exploiting fine-grained task-based execution on multi-GPU systems," in *Proc. IEEE Int. Conf. Cluster Comput.*, Austin, TX, Sep. 2011, pp. 386-394.

[30] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, Jun. 2010, pp. 1-7.

[31] K. D. Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "The energy/frequency convexity rule: Modeling and experimental validation on mobile devices," in *Proc. Springer Int. Conf. Parallel Process. Appl. Mathematics (PPAM)*, Warsaw, Poland, Sep. 2013, pp. 793-803.

[32] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Kluwer J. VLSI Signal Process. Syst.*, vol. 13, no. 2/3, pp. 203-221, Aug. 1996.

[33] T. Cover and J. Thomas, *Elements of Information Theory.* New York, NY, USA: Wiley, 1991.

[34] 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. 3842-3885, Oct. 2015.

[35] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Redondo Beach, CA, Aug. 2012, pp. 279-284.

[36] S. M. Ross, *Introduction to probability models*. Cambridge, MA, USA: Academic Press, 2014.

[37] M. J. Neely, *Stochastic network optimization with application to communication and queueing systems*. San Rafael, CA, USA: Morgan & Calypool, 2010.

[38] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK: Cambridge University Press, 2004.

[39] L. Grippo and M. Sciandron, “On the convergence of the block nonlinear Gauss-Seidel method under convex constraints,” *ELSEVIER Oper. Res. Lett.*, vol. 26, no. 3, pp. 127-136, Apr. 2000.

[40] A. Beck, “On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes,” *SIAM J. Optim.*, vol. 25, no. 1, pp. 185-209, Jan. 2015.