Energy Efficiency and Delay Tradeoff in Multi-User Wireless Powered Mobile-Edge Computing Systems

Sun Mao1, Supeng Leng1*, Kun Yang1†, Quanxin Zhao1, and Ming Liu1

1School of Communication & Information Engineering, University of Electronic Science and Technology of China, Chengdu, China
†School of Computer Science & Electronic Engineering, University of Essex, Colchester, United Kingdom
*Corresponding author, email: spleng@uestc.edu.cn

Abstract—Prolonging battery lifetime and enhancing computation capability have been the key challenges for designing the mobile devices in the Internet of Things (IoT) era. The investigation of Mobile-Edge Computing (MEC) with Wireless Energy Transfer (WET) is a promising solution to overcome such challenges. In this paper, we study the fundamental tradeoff between Energy Efficiency (EE) and delay in the multi-user wireless powered MEC systems. In order to tackle the randomness of channel conditions and task arrivals, we formulate a stochastic optimization problem to achieve the EE-delay tradeoff, which optimizes the network energy efficiency subject to the network stability, Central Processing Unit (CPU)-cycle frequency, peak transmission power, and energy causality constraints. Furthermore, we propose a joint computation allocation and resource management algorithm by transforming the original problem into a series of deterministic optimization problems in each time block based on Lyapunov optimization theory, whose convexity is further proved. Specifically, the proposed algorithm with low complexity requires no prior distribution knowledge of channel conditions and task arrivals. In addition, theoretical analysis shows that the algorithm achieves the EE-delay tradeoff as \[O\left(\frac{1}{V}\right), O(V)\] and provides a control parameter \(V\) to balance the EE-delay performance. Numerical results verify the theoretical analysis and reveal the impacts of various parameters to the system performance.

Index Terms—Mobile-edge computing, wireless energy transfer, energy efficiency, delay.

I. INTRODUCTION

The explosive development of Internet of Things (IoT) has incurred a sharply increase in the computation-intensive mobile application including autonomous driving, virtual reality, interactive online game, etc. The development of these computation-intensive application will be significantly restricted by resource-constraint devices with poor computation capability [1]. Mobile-Edge Computing (MEC) is a promising technique to overcome such challenge, which enables mobile devices to access cloud computing services via integrated edge server at the Access Point (AP) [2]. Therefore, the mobile devices will enjoy better computation experience, e.g., shorter latency, higher bandwidth, access to radio network information, and location awareness, by offloading partial or all of their computation tasks to the edge computing server [3] [4].

On the other hand, conventional IoT is constrained by the limited battery capacity. Finite network lifetime is regarded as another performance bottleneck facing IoT. To remove such bottleneck, many recent research focus on Wireless Energy Transfer (WET), which can provide cost-effective and sustainable energy supply [5] [6]. Introducing WET technology into MEC systems enables to build a novel network paradigm, i.e., wireless powered MEC systems, by which mobile devices can replenish energy from the power beacon and offload partial/all of computation-intensive tasks to the edge server located at the AP.

In the wireless powered MEC systems, the Energy Efficiency (EE) of WET and mobile computation offloading could be low under the poor channel condition. In such cases, we can postpone the energy transfer and...
computation offloading until the channel gets better, however, it may incur large delay. Similarly, the user can conserve the energy for local computation by reducing the Central Processing Unit (CPU) frequency, but at the cost of larger task execution delay. Besides, computation offloading is not a good choice when the amount of computation tasks is particularly large, since transmitting the tasks may consume considerable energy and incur large delay. Motivated by these observations, it is meaningful to balance the EE and delay by jointly scheduling communication, computation and energy resource in the wireless powered MEC systems.

In this paper, we study the inherent EE-delay tradeoff and the resultant joint computation allocation and resource management policy for the newly emerging wireless powered MEC systems. Considering the randomness of channel states and task arrivals, we formulate a stochastic optimization problem to optimize the time-ness of channel states and task arrivals, so that it can be easily applied in practical wireless powered MEC systems. Considering the randomness of channel states and task arrivals, we formulate a stochastic optimization problem to optimize the time-ness of channel states and task arrivals, so that it can be easily applied in practical wireless powered MEC systems.

The main contributions of this paper are summarized as follows:

- We propose an analytical framework to study the fundamental EE-delay tradeoff, and investigate the joint computation allocation and resource management policy for newly multi-user wireless powered MEC systems.
- Based on the Lyapunov optimization technique and convex optimization theory, a joint computation allocation and resource management algorithm is proposed to solve the EE-delay tradeoff problem. In particular, the proposed algorithm with low complexity requires no prior knowledge of channel states and task arrivals, so that it can be easily applied in practical wireless powered MEC systems.
- We prove the EE-delay tradeoff for wireless powered MEC systems as \( O(1/V), O(V) \), where \( V \) is a control parameter. The simulation results validate the tradeoff and show different levels of EE and delay can be achieved by adjusting the control parameter \( V \).

II. SYSTEM MODEL

We consider a multi-user wireless powered Mobile-Edge Computing (MEC) system consisting of an Information Access Point (IAP) (integrated with a MEC server), a power beacon, and \( K \) users denoted as \( U_i \), \( i = 1, \ldots, K \). The IAP and the power beacon have stable energy supply, e.g., powered by grid, whereas each user has no fixed energy supply and needs to harvest energy from the radio signal received from the power beacon. The users utilize the harvested energy to execute their computation tasks by locally computing or (partial) offloading to the IAP.

The time horizon is divided into blocks with equal length \( T \), where the time block \( t \) stands for the time duration \([(t-1)T,tT]\). Each time block, as shown in Fig. 1, consists of three phases, i.e., Wireless Energy Transfer (WET), Computation Task Offloading (CTO), and Cloud Computing and Downloading (CCD). Since the cloud computing has low latency and the downloading is generally faster than offloading, the duration of CCD is negligible and not considered in this paper [7]. The user first receives the energy signal within the duration \( (t_0(t)) \) and then offload partial/all of their computation tasks to the IAP. Let \( h_{D,i}(t) \) and \( h_{U,i}(t) \) denote the downlink and uplink channel gain, respectively. All channels follow quasi-static flat-fading, where the channel state remains constant during each time block, but possibly varying at the boundary of the time block [8] [9]. We also assume that the Channel State Information (CSI) is perfectly known at the power beacon.

A. Task Offloading Model

The user \( i \) computes \( D_{i,t}(t) \)-bit task locally and offloads \( D_{o,t}(t) \)-bit task to the IAP within the time block \( t \). Let \( N_i \) denote the number of central processing unit (CPU) cycles required to process one bit task for \( U_i \), which depends on the types of applications. The CPU-cycle frequency scheduled for \( U_i \) in the \( t \)-th time block is denoted as \( f_i(t) \). Thus, the computation tasks
processed via Local Computing (LC) is given by
\[ D_{l,i}(t) = \frac{T f_i(t)}{N_i}, \tag{1} \]

Denote the uplink transmission power at \( U_i \) as \( P_i^U(t) \), \( i = 1, \cdots, K \), hence, the amount of offloaded tasks by \( U_i \) in time slot \( t \) is
\[ D_{r,i}(t) = B R_i(t) \log_2(1 + \frac{P_i^U(t)}{\delta^2}), \tag{2} \]
where \( B \) denotes the bandwidth for offloading and \( \delta^2 \) the noise power.

Let \( D_i(t) \) denote the total accomplished computation tasks of \( U_i \) in time block \( t \), we have
\[ D_i(t) = D_{l,i}(t) + D_{r,i}(t). \tag{3} \]

Each user equips with a task buffer and a battery. Let \( A_i(t) \) denote the amount of computation task arrived at user \( i \) at the end of the time block \( t \). Without loss of generality, we assume that \( A_i(t) \) is independent and identically distributed over all the time blocks with the average arrival rate \( \mathbb{E}[A_i(t)] = \lambda_i \), hence, the queue length of the task buffer \( Q^D_i(t) \) evolves according to the following equation
\[ Q^D_i(t + 1) = [Q^D_i(t) - D_i(t)]^+ + A_i(t), \tag{4} \]
where \([x]^+ = \max(x, 0)\).

**B. Energy Consumption Model**

Let \( P_0^D(t) \) denote the transmission power by the power beacon during \( \tau_0(t) \), hence, the energy harvested by \( U_i \) during the \( t \)-th time block can be expressed by
\[ E_{H,i}(t) = \eta_i h_{D,i}(t) P_0^D(t) \tau_0(t), \tag{5} \]
where \( \eta_i \in (0, 1) \) represents the efficiency of energy harvesting at \( U_i \).

The total energy consumption of the proposed MEC systems is divided into two parts, i.e., the service energy consumption, and the total user energy consumption. Let \( E_S(t) \) denote the service energy consumption. It is further divided into two parts: one part for combating the path loss and channel fading during the downlink energy transmission stage, and the other part is the computation energy consumption of the MEC server, hence
\[ E_S(t) = P_0^L(t) \tau_0(t) - \sum_{i=1}^{K} E_{H,i}(t) + E_{C,0}. \tag{6} \]
where \( E_{C,0} \) denote the computation energy consumption of the server. Since the server has strong computation capacity, we assume that it consumes constant energy \( E_{C,0} \) to minimize the computation time.

The energy consumption at \( U_i \) consisting of two parts, i.e., the uplink transmission consumption and the local CPU energy consumption. According to [2], the local computation consumption is expressed as
\[ E_{C,i}(t) = \kappa T f_i^3(t), \tag{7} \]
where \( \kappa \) is the effective switched capacitance that depends on the chip architecture [2]. The total energy consumption at \( U_i \) is
\[ E_{U,i}(t) = E_{C,i}(t) + P_i^U(t) \tau_i(t). \tag{8} \]
Therefore, the energy stored in the battery of \( U_i \) is updated as
\[ Q_i^E(t + 1) = [Q_i^E(t) - E_{U,i}(t)]^+ + E_{H,i}(t). \tag{9} \]
Limited by the energy causality constraint, hence, the energy consumption by \( U_i \) is no more than current available energy in the battery, hence
\[ E_{U,i}(t) \leq Q_i^E(t) + E_{H,i}(t). \tag{10} \]
Accordingly, the total energy consumption of the system in time block \( t \) is given by
\[ E_{TOT}(t) = E_S(t) + \sum_{i=1}^{K} E_{U,i}(t). \tag{11} \]

**III. Problem Formulation**

In this section, we first introduce the definition of energy-efficiency (EE). Then, we formulate the EE-delay tradeoff problem for the proposed systems.

**A. Definition of EE**

The network EE is defined as the ratio of long-term total energy consumption to the corresponding long-term aggregate accomplished computation tasks
\[ \eta_{EE} = \lim_{N \to \infty} \frac{1}{N} \frac{1}{N-1} \sum_{t=0}^{N-1} \mathbb{E}\{E_{TOT}(t)\} = \frac{\mathcal{E}_{TOT}}{K \sum \mathcal{T}_i}, \tag{12} \]
where \( \mathcal{E}_{TOT} \) and \( \mathcal{T}_i \) denote the limit of time average expectation of \( E_{TOT}(t) \) and \( D_i(t) \), respectively.

**B. EE-Delay Tradeoff Problem**

In order to reveal the fundamental tradeoff between the EE and delay in the considered systems, we introduce a problem to minimize the network EE subject to the network stability constraint, which can be expressed as
\[ \min_{\tau(t),P^U(t),f(t)} \eta_{EE} \]
In order to solve the problem (13) by Lyapunov optimization, we first define the quadratic Lyapunov function as

$$L(t) = \frac{1}{2} \sum_{i=1}^{K} (Q_i^D(t))^2.$$  \hfill (16)

Accordingly, we further define two important functions, i.e., Lyapunov drift function and Lyapunov drift-plus-penalty function, as

$$\Delta L(t) = \mathbb{E}\{L(t+1) - L(t)\mid Q_i^D(t)\},$$

$$\Delta_V L(t) = \Delta L(t) + V \mathbb{E}\{E_{TOT}(t) - \eta_{EE}(t) \sum_{i=1}^{K} D_i(t)\mid Q_i^D(t)\},$$ \hfill (17)

where $V$ is a control parameter to achieve EE-delay tradeoff, and $Q_i^D(t) = (Q_1^D(t), Q_2^D(t), \ldots, Q_K^D(t))$. Obviously, minimizing the drift-plus-penalty function $\Delta_V L(t)$ can carry two targets, i.e., ensuring network stability and minimizing network EE. Furthermore, an upper bound of $\Delta_V L(t)$ is derived to simplify the problem.

**Lemma 1:** The Lyapunov drift-plus-penalty follows the equation below,

$$\Delta_V L(t) \leq C - \sum_{i=1}^{K} Q_i^D(t) \mathbb{E}\{D_i(t)\mid Q_i^D(t)\},$$

$$+ V \mathbb{E}\{E_{TOT}(t) - \eta_{EE}(t) \sum_{i=1}^{K} D_i(t)\mid Q_i^D(t)\},$$ \hfill (18)

where $C$ is a bounded constant.

**Proof:** Proof is omitted due to space limitation.

Therefore, we can minimize the right side of inequality in Eq. (18) to achieve the EE-delay tradeoff. According to the principle of opportunistically minimizing an expectation [11], we can solve the following optimization in each time block to acquire the optimal solution of the original problem (13) [11].

$$\min_{\tau(t), P_i^T(t), f(t)} V(E_{TOT}(t) - \eta_{EE}(t) \sum_{i=1}^{K} D_i(t))$$

$$- \sum_{i=1}^{K} Q_i^D(t) D_i(t)$$

s.t. \hspace{1cm} C1-C6. \hfill (19)

Due to the non-convexity of energy casualty constraint C5, it is difficult to use the classic convex optimization to find a optimal solution. We transform the problem (19) to a convex problem (20) by introducing a set of auxiliary variables, i.e., $\varepsilon_i^T(t) = P_i^T(t) \tau_i(t), i =
Algorithm 1: Lyapunov Optimization-Based Joint Computation Allocation and Resource Management Policy

1. At each time block $t$, obtain $\{Q_D(t)\}$, $\{Q_E(t)\}$, $\{h_{D,i}(t)\}$, $\{h_{U,i}(t)\}$, and $\eta_{EE}(t)$.
2. Solve the convex problem (21) to acquire the optimal computation allocation and resource management policy.
3. Update $\{Q_D(t)\}$, $\{Q_E(t)\}$, and $\eta_{EE}(t)$ according to Eqs. (4), (9), and (15), respectively.
4. Repeat steps 1-3.

C. Performance Analysis

In this subsection, we derive the gap between the optimal EE acquired by Algorithm 1 and the optimal value of the original problem (13), and deduce queue length boundary of task buffer. Also, the EE-delay tradeoff will be revealed.

Theorem 2: Suppose the problem (13) is feasible, we have

- The network EE obtained by Algorithm 1 is upper bounded by

$$
\eta_{EE} \leq \eta_{EE}^{opt} + \frac{C}{V \sum_{i=1}^{K} D_{i,\min}},
$$

(21)

- The time-average sum data queue length is bounded by

$$
\overline{Q}^D \leq \frac{C + V(\eta_{EE}^{opt} \sum_{i=1}^{K} D_{i,max} - E_{TOT,\min})}{\epsilon}
$$

(22)

Proof: Proof is omitted due to space limitation.

V. NUMERICAL RESULTS

In this section, elaborate simulation results are provided to validate the theory analysis and to evaluate performance of the proposed joint computation allocation and resource management policy. In the simulations, all the channels follow the i.i.d. Rayleigh fading, where the channel power gains are subject to the Gamma distribution. The mean value of $h_{U,i}$ and $h_{D,i}$ are $0.1 \times i$ and $0.2 \times i$, respectively. The noise power satisfies $\delta^2 = 10^{-6}$ W and the bandwidth is $B = 5$ KHz. In addition, the peak transmission power at the power beacon is set as $P_{\text{max}} = 4$ W and the time duration satisfies $T = 100$ ms. For user $i$, we set $\eta_i = 0.7$, $\kappa = 10^{-28}$, $N_i = 788$ cycles/bit, $P_{\text{max}} = 0.2$ W [13].

In Figs. 2-3, we show the relationship between the energy efficiency/average queue length and the control parameter $V$. As can be seen, the energy efficiency decreases as $V$ increases, and converges to the optimal value $\eta_{EE}^{opt}$ when $V$ is sufficiently large. In addition, the average queue length increases linearly with the value of the control parameter $V$. From Figs. 2-3, we can see that the energy efficiency and delay increase with the workload of the network (task arrival rate $\lambda_i(t)$ and user number $K$). Obviously, a heavier workload will lead to the increase of the queue backlog. Meanwhile,
more energy is consumed to keep the queue stability, and the accomplished tasks increase in a slower rate compared to the energy consumption according to Eqs. 1-2, 7-8, leading to the increase of the network energy efficiency.

Fig. 4 shows the relationship between the energy efficiency and the average execution delay. It is easily observed that the average execution delay increases as the network energy efficiency decreases. The obtained results in Figs. 2-4 verify the $[O(1/V), O(V)]$ tradeoff between energy efficiency and delay, which is described in Theorem 2. Therefore, a proper $V$ should be chosen to balance two desirable objectives. If we aim to improve network EE, a larger $V$ can be chosen. On the contrary, a smaller $V$ will be appropriate.

VI. CONCLUSION

This paper investigated the fundamental EE-delay tradeoff in multi-user wireless powered mobile-edge computing systems. In order to reveal the EE-delay tradeoff, a stochastic optimization problem was formulated, which optimizes the network EE subject to the energy causality and network stability constraints. Based on the Lyapunov optimization method and convex optimization theory, we proposed a joint computation allocation and resource management algorithm to solve the energy efficiency-delay tradeoff problem by transforming the original problem to a series of deterministic optimization problems in each time block. The proposed algorithm with low complexity since it requires no prior knowledge of channel conditions and task arrivals. We further analyzed the performance of our proposed algorithm, which indicates that the network energy efficiency and delay obeys an $[O(1/V), O(V)]$ tradeoff with $V$ as a control parameter. Simulation results validated the tradeoff and revealed the impacts of various parameters to the system performance.

REFERENCES