sensors

MDPI

Article

Energy-Efficient Online Resource Management and Allocation Optimization in Multi-User Multi-Task Mobile-Edge Computing Systems with Hybrid Energy Harvesting

Heng Zhang [1,2,†], **Zhigang Chen** [1,2,*,†], **Jia Wu** [1,2,*,†], **Yiqing Deng** [1,2,†], **Yutong Xiao** [1,2,†], **Kanghuai Liu** [1,2,†] **and Mingxuan Li** [1,2,†]

[1] School of Software, Central South University, Changsha 410075, China; zhangheng1018@csu.edu.cn (H.Z.); dengyiqin@csu.edu.cn (Y.D.); tdxytcn@gmail.com (Y.X.); Liukanghuai@csu.edu.cn (K.L.); LiMingXuan@csu.edu.cn (M.L.)

[2] "Mobile Health" Ministry of Education-China Mobile Joint Laboratory, Changsha 410075, China

* Correspondence: czg@csu.edu.cn (Z.C.); jiawu0510@csu.edu.cn (J.W.); Tel.: +86-133-8748-0797 (Z.C.)

† These authors contributed equally to this work.

Received: 25 July 2018; Accepted: 14 September 2018; Published: 17 September 2018

Abstract: Mobile Edge Computing (MEC) has evolved into a promising technology that can relieve computing pressure on wireless devices (WDs) in the Internet of Things (IoT) by offloading computation tasks to the MEC server. Resource management and allocation are challenging because of the unpredictability of task arrival, wireless channel status and energy consumption. To address such a challenge, in this paper, we provide an energy-efficient joint resource management and allocation (ECM-RMA) policy to reduce time-averaged energy consumption in a multi-user multi-task MEC system with hybrid energy harvested WDs. We first formulate the time-averaged energy consumption minimization problem while the MEC system satisfied both the data queue stability constraint and energy queue stability constraint. To solve the stochastic optimization problem, we turn the problem into two deterministic sub-problems, which can be easily solved by convex optimization technique and linear programming technique. Correspondingly, we propose the ECM-RMA algorithm that does not require priori knowledge of stochastic processes such as channel states, data arrivals and green energy harvesting. Most importantly, the proposed algorithm achieves the energy consumption-delay trade-off as $[\mathcal{O}(1/V), \mathcal{O}(V)]$. V, as a non-negative weight, which can effectively control the energy consumption-delay performance. Finally, simulation results verify the correctness of the theoretical analysis and the effectiveness of the proposed algorithm.

Keywords: mobile edge computing; Internet of Things; hybrid energy harvesting; Lyapunov optimization; energy efficiency; delay

1. Introduction

As an ubiquitous computer paradigm, the Internet of Things (IoT) has seen explosive growth in computationally intensive mobile applications such as autonomous driving, virtual reality, and interactive online games [1]. Because of size and production cost considerations, wireless devices (WDs) (e.g., sensors) in IoT systems generally carry capacity-constrained batteries and energy-saving, low-performance processors. Therefore, the development of computationally intensive applications is severely limited by resource-constrained devices [2,3]. Consequently, how to eliminate this bottleneck is a key issue in the research and development of modern Internet of Things technology.

Currently, this limitation is solved by transferring computation and storage from the resource-constrained devices to clouds. However, cloud computing resources are deployed in large data centers far from most users, resulting in communication delay and high energy consumption between users and clouds [4]. In order to solve these problems, the concept of edge and fog computing was proposed [5]. MEC (Mobile Edge Computing)is a promising technology that can overcome these challenges and allows WDs to access cloud computing services on the access point (AP) and on base station (BS) integrated the edge servers. In the MEC system, resource-constrained WDs can offload computation tasks to APs and then these tasks computed by MEC servers are deployed on the APs [6]. The edge computing platform differs from cloud computing in that the computing resources are at the edge of the network and are near the end point device. Therefore, it is possible for MEC to reduce communication delay and the bandwidth requirement of the network connection to the remote data center. In addition, based on the convenience of data processing and storage near users, some new research contents will be generated [5,7]. At present, there are many studies on the direction of computation offloading in the field of MEC.

The computation offloading technology in the MEC can effectively enhance the computation capability of WDs, but how can we provide a sustainable, stable and effective energy supply for WDs? Energy harvesting (EH) technology is emerging as a promising paradigm to solve the limited battery capacity problem. Currently, green energy has been widely used in the industry as an environmentally friendly renewable energy source [8–10]. There are many benefits to using green energy, but there are some problems with low energy conversion rate tightly correlated to the environmental conditions. Consequently, green energy is highly unpredictable and has instability [11]. To tackle this issue, we introduce radio frequency (RF) based on wireless power transfer (WPT). WPT can use RF energy transmitters to continuously charge the batteries of remote energy harvesting IoT devices [2,12,13]. Therefore, we use green energy supply as a primary energy source of WDs, and the backup energy source supply is wireless energy based on WPT.

In order to improve the energy efficiency and performance of WDs in an MEC system, in addition to designing an efficient computational offloading scheme, we need to consider the following issues: (i) How much transmit power should be allocated to offloading computation tasks; (ii) How long to offload tasks; and (iii) how to manage the battery of an WD to ensure the normal operation of the device? In order to solve the above problems, we focus on the joint resource management and allocation issues. The main contributions are as follows:

- We consider a multi-user multi-task MEC system with hybrid energy harvesting WDs to investigate the joint resource management and allocation problem. To solve the stochastic optimization problem, we turn the problem into two deterministic sub-problems, namely transmission optimization sub-problem and battery management sub-problems. The transmission optimization sub-problem can be designed to solve the transmission power and the transmission time of data, while the battery management sub-problem can be constructed to solve the needed wireless energy amount.
- Based on Lyapunov optimization theory and convex optimization theory, we use the proposed energy-efficient joint resource management and allocation algorithm (ECM-RMA) to solve the problem of minimizing energy consumption under delay guarantees. Without the prior knowledge, the algorithm allocates energy offloading data according to the energy harvest amount, remaining available time, amount of data to be processed and other factors at the beginning of each time slot.
- The trade-off between energy consumption and delay is $[\mathcal{O}(1/V), \mathcal{O}(V)]$. V, as a non-negative weight, can achieve the balance between energy consumption and task data queues.
- In the field of MEC, we first proposed the devices with a hybrid energy harvesting method that integrates green energy and wireless energy.

2. Related Works

Recently, the academic community has done a lot of research on resource management and allocation strategies in an MEC system. These policies mainly manage and distribute resources of local execution, comunication process, and edge servers. The scenario of resource management and allocation in MEC systems is divided into single-user MEC systems [14–19], multi-user MEC systems [20–26] and heterogeneous server MEC systems [27,28]. In [14], the offloading ratio, transmission power, and the CPU clock frequency are jointly optimized to minimize the latency subject to the energy consumption or minimize the energy consumption subject to the latency. In [19], Mahmoodi jointly optimizes the computational latency and energy consumption, which only based on the Markov decision processes (MDP) theory to optimize the size of the offloading data. In order to minimize the total energy consumption, an optimization strategy is proposed in [20], which has a simple threshold structure for the size of offloading data and time allocation, and is controlled by the optimization strategy. In [21], instead of controlling the size of the offloading data and the time allocation, Barbarossa minimizes the overall energy consumption by controlling the transmission power and the server CPU clock. However, the above works did not consider a real application scenario, which is resource management and allocation under multi-user and multi-task.

In order to prolong the battery life of wireless IoT devices and to reduce its manual maintenance costs, some recent works have begun to explore renewable energy supplies and wireless transfer energy supplies in MEC systems. In [29,30], researchers incorporate renewable energy into MEC. In [29], an effective resource management algorithm based on reinforcement learning is proposed by Xu and Ren. Compared with standard reinforcement learning algorithms such as Q-learing, learning speed and time performance of the proposed effective resource management algorithms are significantly improved. Maos. proposed a low-complexity online algorithm to jointly determine the offloading strategy, CPU cycle frequency and transmission power to achieve an asymptotically optimal of execution cost in [30]. In wireless powered MEC scenes, an optimized resource allocation strategy in [6] is proposed by Wang, and the total energy consumption of the AP is minimized under the user's own computational delay constraint. In addition, in a multi-user wireless powered MEC network, in order to maximize the weighted total calculation rate, an alternating direction method of multipliers (ADMM) -based technique by jointly optimizing the computational model and transmission time allocation was proposed in [2]. However, the strong randomness of renewable energy cannot guarantee the reliability of local execution or offloading. In addition, in the wireless transfer energy powered MEC system, it will cause some energy consumption to the energy transmission end. To the best of our knowledge, it has not been proposed that the devices in an MEC system use a hybrid energy harvesting method for local computing and offloading.

In addition, the number of available sub-channels in the IoT network also have an larger impact on tasks offloading. In [1], Mao use a joint computation allocation and resource management algorithm to research the trade-off of fundamental between Energy Efficiency and delay. In order to minimize energy consumption of device while meeting delay constarint of user, Hao et al. propose a joint task offloading and caching problem in [31]. An energy-efficient dynamic offloading resource scheduling (eDors) policy is proposed by Guo to shorten task completion time and reduce energy consumption [25]. However, the above works did not consider the number of available sub-channels in an MEC system.

The remainder of this paper is as follows: Section 3 shows the system model and the problem of minimizing the average total energy consumption subject to latency is formulated. Section 4 proposes an efficient online resource management and allocation algorithm to solve the formulated problem and the performance analysis of the algorithm is given. Section 5 provides simulation results to verify the correctness of the theoretical analysis and the effectiveness of the proposed algorithm. Finally, Section 6 concludes this paper. The key mathematical notations used in this paper are listed in Table 1, and the abbreviations used in the Table 1 are defined in Table 2.

Table 1. Definition of notations.

Notation	Defination
$\mathcal{M}$	The set of WDs
$\mathcal{N}$	The set of tasks
$\mathcal{T}$	The set of time slots
$p_m(t)$	The uplink transmission power between WD m and the AP at time slot t
$h_{m,ul}(t)$	The uplink channel gains between WD m and the AP at time slot t
w_0	The white noise power level of uplink channel
$B_{m,ul}$	The uplink channel bandwidth that the AP allocates for the WD m at time slot t
$v_m(t)$	The uplink transmission rate between WD m and the AP
$\tau_m^n(t)$	The duration of task n partially offloaded from WD m to the AP within time slot t
$\tau_m(t)$	The time allocated schedule of all tasks in WD m during slot t
$I_{\mathrm{m}}^n(t)$	The amount of computation task arrived at user m at the end of the time slot t
$f_m^n(t)$	The CPU-cycle frequency assigned to task n on WD m at time slot t
$L_m(t)$	The CPU-cycles required to calculate one bit of data in WD m at time slot t
$D_{m,l}^n(t)$	The local computed data amount of task n of WD m at time slot t
$e_{m,l}^n(t)$	The energy consumption for local computation for task n of WD m during a time slot t
$D_{m,c}^n(t)$	The offloaded data amount of task n of WD m at time slot t
$e_{m,c}^n(t)$	The energy consumption of WD m offloading the data of task n at time slot t
$D_m^n(t)$	The total computed data of task n of WD m at time slot t
$Q_m^n(t)$	The backlog of the data queue for task n of WD m at time slot t
$e_{m,h}(t)$	The energy obtained from the environment of WD m at time slot t
$e_{m,w}(t)$	The energy obtained from the AP of WD m at time slot t
$e_{m,hw}(t)$	The total harvested energy of WD m at time slot t
$\gamma(t)$	The energy supply from the environment at time slot t
$e_{m,total}^n(t)$	The total energy consumed by WD m for task n within time slot t
$E_m(t)$	The dynamic change of WD's energy queue
M	The number of WDs
N	The number of tasks in a WD
T	The slot length
S	The number of slots
K	The number of available sub-channel
$p_{m,max}$	The maximum uplink transmission power
$p_{m,min}$	The minimum uplink transmission power
$I_m^{n,max}$	The maximum arrive data
γ_{max}	The maximum green energy supply

Table 2. Definition of abbreviations used in the Table 1.

Abbreviation	Defination
WD	Wireless device
WDs	Wireless devices
AP	Access point

3. System Model

As shown in Figure 1, we consider a multi-user multi-task MEC system with hybrid energy harvesting, which consists of an AP (integrated with an RF energy transmitter, an MEC server and a communication circuit) and M WDs (each WD contains an energy harvesting circuit, a rechargeable battery, a computing unit and a communication circuit). We denote the set of WDs by $\mathcal{M}$ = {1, 2, ..., M} and use $\mathcal{N}$ = {1, 2, ..., N} to denote the set of computation tasks of WD m. Each WD has a completion deadline, and J_m denotes the completion deadline of WD m. Moreover, the number of available sub-channels for the WDs is denoted by K. We divide the total time into slots denoted by set of $\mathcal{T}$ = {0, 1, ...,} and $t \in \mathcal{T}$ with slot length T. Within each time frame, some computationally intensive tasks can be offloaded to the AP through uplink wireless links and computed at the MEC server, while others are directly computed locally.

We assume that the AP knows the channel status information (CSI), task offloading information, and the remaining battery power of WDs, which can be obtained through feedback [20]. Using this information, at the beginning of slot t, the data transmission power, the transmission time and the energy need to be harvested by each device will be determined by the AP. In addition, then the AP sends these decisions to the WDs in the MEC system. Specially, we only need to focused on one time slot.

From Figure 2, the WDs in a hybrid energy harvesting MEC system powered by green energy from the environment and wireless transfer energy from the AP as well as the AP powered by green energy and grid energy can be seen. In detail, when a device cannot get enough energy from the environment to fill the rechargeable battery, AP in the MEC system will transmit wireless energy to fill the device battery. In addition, for the AP, green energy is used as the primary source of energy, while stable grid energy as a backup energy supply [2]. Then, the harvested energy of a WD is used by the computing unit and the communication circuit to compute tasks locally and offload tasks, respectively.

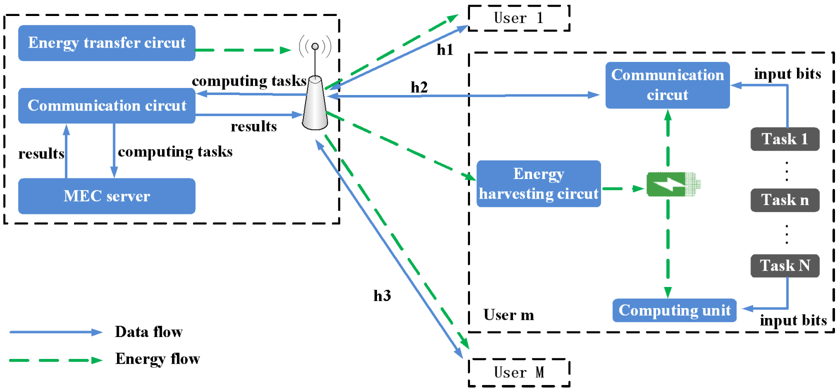


Figure 1. The detailed multi-user multi-task MEC (Mobile Edge Computing) system architecture when only considering IoT (Internet of Things) devices with wireless powered model.

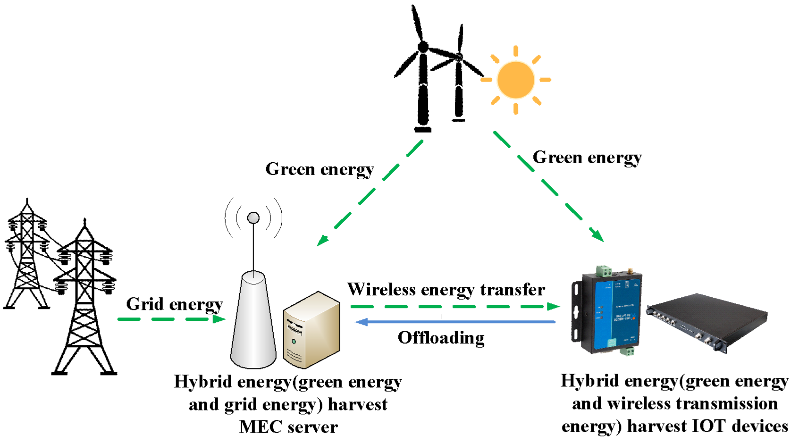

Green energy
Green energy
Grid energy
Wireless energy transfer
Offloading
Hybrid energy(green energy and grid energy) harvest MEC server
Hybrid energy(green energy and wireless transmission energy) harvest IOT devices

Figure 2. A multi-user multi-task MEC system with hybrid energy harvesting.

3.1. Communication Model

We first introduce the communication model between the WDs and the AP in an MEC system. Small-scale Rayleigh fading is followed by all wireless channels. Therefore, the wireless channel states are independent and remain unchanged during each time slot. During a time slot t, the uplink transmission rate between WD m and the AP is quantified as

$$\nu_m(t) \; = \; B_{m,ul}\log_2(1+\frac{p_m(t)h_{m,ul}(t)}{w_0}), \tag{1}$$

where $B_{m,ul}$ is the uplink channel bandwidth, which is allocated to WD m by the AP. $p_m(t)$ represents the transmission power between WD m and the AP at time slot t. Moreover, $p_m(t)$ with the minimum value $p_{m,min}$ and the maximum value $p_{m,max}$, i.e.,

$$p_{m,min} \leq p_m(t) \leq p_{m,max}. \tag{2}$$

$h_{m,ul}(t)$ is the uplink channel gain between WD m and the AP. Specially, the channel gain is determined by small-scale Rayleigh fading and the distance-dependent path loss. In addition, w_0 is the white noise power level.

At the same time, a WD can only occupy one available sub-channel. Let $\tau_m^n(t)$ denote the duration of task n partially offloaded from WD m to the AP within time slot t, i.e.,

$$0 \leq \tau_m^n(t) \leq T. \tag{3}$$

Moreover, we define $\tau_m(t) = \{\tau_m^1(t), \tau_m^2(t), ..., \tau_m^N(t)\}$ as the time allocated schedule of all tasks in WD m during slot t, and the total uplink transmission time of WD m must not exceed the available slot length, given as

$$\sum_{n\in\mathcal{N}} \tau_m^n(t) \leq T. \tag{4}$$

In addition, the total uplink transmission time of the all tasks on all devices must not exceed the sum of the time lengths of all available sub-channels, which is as follows:

$$\sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}} \tau_m^n(t) \leq KT, \tag{5}$$

where K denotes the number of available sub-channels in the MEC system, and each WD can only access an available sub-channel at one time slot due to the WDs in a IoT network being narrow-band and simple.

In this paper, we don't consider the delay, energy consumption and packet loss of downlink transmission. This is because the size of the result returned is normally much smaller than the corresponding pre-processing task. In addition, the downlink transmission rate between the WDs and the AP is higher than the corresponding uplink transmission rate [6,31].

3.2. Task Computation Model

In this section, we introduce the task computation model. In practice, many computed tasks consist of multiple small procedures/components. Thus, it is necessary to perform partial computation offloading of a task. Specifically, a computed task can be divided into multiple parts where some of the parts were executed at the WD and the others were offloaded for the AP execution. Let $I_m^n(t)$ denote the data amount of computed task n that arrived at WD m at the end of time slot t and the WD can continuously receive data within a time slot. $I_m^n(t)$ with the rate of data arrival $C_m^n(t)$, which obeys independently and identically distributed (i.i.d) over time slots, independent between tasks, have a maximum data arrival rate $C_m^{n,max}(t)$ and have average arrival data rate λ_m^n. Therefore, $I_m^n(t)$ has the maximum data arrival $I_m^{n,max}$, i.e.,

$$0 \leq I_m^n(t) \leq I_m^{n,max}. \tag{6}$$

Moreover, each WD allocates the same amount of data storage space R^D for each task, and the task data is stored in its own corresponding data storage space before it is processed.

3.2.1. Local Computing

Regarding the local computing, let $D^n_{m,l}$ denote the data amount of task n computed locally by the WD m within a time slot t. The CPU-cycle frequency assigned to task n in WD m is denoted by f^n_m, and L_m is the CPU cycles required to compute a one bit size of data. Therefore, the local computed data amount of task n in WD m during time slot t can be expressed as follows:

$$D^n_{m,l} = T\frac{f^n_m}{L_m}. \tag{7}$$

The corresponding energy consumption for local computation for task n in WD m during a time slot t is given as follows:

$$e^n_{m,l}(t) = \kappa[f^n_m]^3 T, \tag{8}$$

where k is the effective switched capacitance related to the chip architecture [1].

3.2.2. Mobile Edge Cloud Computing

On the other hand, because of WDs not having enough computing power, some data need to be offloaded to the AP for computation. Our paper is aimed at the energy consumption of the WDs in an MEC system, so we do not consider the energy consumption generated by the MEC server.

We assume that the offloaded data amount of task n in WD m at time slot t is shown by

$$D^n_{m,c}(t) = \nu_m(t)\tau^n_m(t). \tag{9}$$

In addition the corresponding energy consumption of WD m offloading the data of task n at time slot t can written as

$$e^n_{m,c}(t) = p_m(t)\tau^n_m(t). \tag{10}$$

After a task is offloaded to the MEC server, the MEC server will process the task. We assume that the computation capability of the MEC server is f_{ap}, and the computing resource required for task n of WD m is g^n_m. Hence, the computation delay generated by processing task n of WD m by the MEC server can be expressed as

$$\tau^{m,n}_{ap} = \frac{g^n_m}{f_{ap}}. \tag{11}$$

As mentioned before, each WD has a completion deadline. Thus, we have

$$\sum_{n\in\mathcal{N}} \tau^{m,n}_{ap} + \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} \tau^n_m(t) \le J_m. \tag{12}$$

Clearly, we have the total computed data of task n in WD m at time slot t, $D_{m,n}(t)$, as given by

$$D^n_m(t) = D^n_{m,l}(t) + D^n_{m,c}(t). \tag{13}$$

In this paper, each task has a data backlog queue. Hence, we use $Q^n_m(t)$ to denote the backlog of the data queue at task n of WD m and it is updated over time. Based on the above analysis, the dynamics of data backlog queue at task n of WD m can easily be expressed as

$$Q^n_m(t+1) = max[Q^n_m(t) - D^n_m(t), 0] + I^n_m(t). \tag{14}$$

3.3. Energy Consumption Model

In this section, we will introduce the energy consumption model. As we know, green energy is unpredictable and unstable. In order to be environmentally friendly and improve the reliability of energy harvesting as much as possible, we use two energy sources to support the WDs in an MEC system. The one energy source is the green energy that WDs harvested from the environment, and the other one is the wireless transfer energy from the AP. In our paper, the AP will provide wireless transfer energy to support the WD when green energy can't support the power required by a WD.

In a given time slot, the energy that WD m gets from the environment is denoted as $e_{m,h}$, which is a stochastic value determined by the state of environment, and $e_{m,w}$ is defined as the energy that WD m gets from the AP. Moreover, the energy supply from the environment is denoted by $\gamma_m(t)$, that is, the green energy that WD m can be get at time slot t. $\gamma_m(t)$ is i.i.d in different time slots, and the upper limitation of $\gamma_m(t)$ is γ_{max}. Since the harvested energy from the environment by any WD cannot exceed the energy supply in a time slot, the following inequality can be obtained:

$$0 \leq e_{m,h}(t) \leq \gamma_m(t), \forall m \in \mathcal{M}. \tag{15}$$

In addition, $e_{m,w}$ is defined as

$$e_{m,w}(t) = \mu p_{ap}(t) h_{m,dl}(t) \ell_m(t), \tag{16}$$

where $\mu \in (0,1)$ represents the energy harvesting efficiency of energy harvesting from an AP. $p_{ap}(t)$ and $h_{m,dl}(t)$ denote the energy transmission power of the AP and the downlink channel gain, respectively. As for $\ell_m(t)$, it is the transmission time of wireless energy, and it satisfies $\ell_m(t) \leq T$. In order to avoid mutual interference caused by the common use of channels, wireless energy transfer and task offloading cannot be performed simultaneously. Thus, we use the following formula to limit:

$$\sum_{n \in \mathcal{N}} \tau_m^n(t) + \ell_m(t) \leq T. \tag{17}$$

Similarly, convert Equation (5) to Equation (17)

$$\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \tau_m^n(t) + \sum_{m \in \mathcal{M}} \ell_m(t) \leq KT. \tag{18}$$

Therefore, the total energy harvested by WD m at time slot t is $e_{m,hw}(t)$, i.e.,

$$e_{m,hw}(t) = e_{m,h}(t) + e_{m,w}(t), \forall m \in \mathcal{M}. \tag{19}$$

As we mentioned before, energy harvested by each WD is stored in a rechargeable battery. The energy queue of WD m is denoted by $E_m(t)$, which is energy available in the battery of WD m. Within time slot t, the total energy consumed by WD m for task n is expressed as $e_{m,total}^n(t)$, which consists of two parts, the local CPU energy consumption and the data offloaded energy consumption, i.e.,

$$e_{m,total}^n(t) = e_{m,l}^n(t) + e_{m,c}^n(t), \forall m \in \mathcal{M}. \tag{20}$$

Based on the above analysis, the dynamic energy queue of WD m can be expressed as

$$E_m(t+1) = max[E_m(t) - \sum_{n \in \mathcal{N}} e_{m,total}^n(t), 0] + e_{m,hw}(t), \tag{21}$$

where $\sum_{n \in \mathcal{N}} e_{m,total}^n(t)$ satisfies the following inequality, i.e.,

$$\sum_{n\in\mathcal{N}} e^{n}_{m,total}(t) \leq E_m(t), \tag{22}$$

since the energy consumption of the WD m cannot exceed the length of its energy queue. Moreover, due to the limited battery capacity of the WD, the sum of the available energy and the harvested energy of the WD cannot exceed the capacity of the battery, which is shown as

$$E_m(t) + e_{m,hw}(t) \leq \Omega, \forall m \in \mathcal{M}. \tag{23}$$

As for the total energy consumption of the WDs in the MEC system at time slot t, it is defined as

$$e(t) = \sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}} e^{n}_{m,total}(t). \tag{24}$$

3.4. Optimization Problem Formulation

Based on the above model, we turn the energy minimization problem into a stochastic optimization programming problem. Especially, with the help of Lyapunov optimization theory, we turn the stochastic optimization problem $\boldsymbol{P1}$ into a series of deterministic optimization problems, where each problem is processed in each time slot and is hoped to be solved through the standard convex optimization technology. Thus, the joint optimization of data transmission power, transmission time and need to harvest energy is made to minimize the total energy consumption of the WDs in the MEC system.

First of all, it can be learned from the previous section that the total energy consumption of the WDs in an MEC system consists of the local CPU energy consumption and the data offloaded energy consumption. Now, we denote the average time energy consumption of the WDs in an MEC system as

$$\bar{e} = \lim_{S\to+\infty} \frac{1}{S}\mathbb{E}[\sum_{t=0}^{S-1} e(t)], \tag{25}$$

where S is the total time length of the MEC system running.

Secondly, we model the problem of minimizing energy consumption. In order to simplify the description, we use $\boldsymbol{p}(t)$, $\boldsymbol{\tau}(t)$ and $\boldsymbol{e}(t)$ to represent the set of transmission power $p_m(t)$, transmission time $\tau_m(t)$ and amount of wireless harvested energy $e_{m,w}(t)$ at time slot t. Moreover, we use $\boldsymbol{\xi}(t) = (\boldsymbol{p}(t), \boldsymbol{\tau}(t), \boldsymbol{e}(t))$ to denote the set of all variables that need to be optimized at time slot t. Therefore, the mathematical problem of minimizing energy consumption can be modeled as follows:

$$\begin{aligned} &\boldsymbol{P1} : \underset{\xi(t)}{min}\, \bar{e}, \\ &s.t. (2)\text{–}(6), (12), (15), (17), (18), (22), (23). \end{aligned} \tag{26}$$

In the next section, we will give a detailed method to solve problem $\boldsymbol{P1}$.

4. Online Resource Management and Allocation Optimization in MEC

In this section, in order to solve the problem of queue stability constraints $\boldsymbol{P1}$, we use Lyapunov theory to decompose the problem into two sub-problems in a single time slot, and give the energy minimization framework. Based on this framework, an algorithm that guarantees the stability of data queues and energy queues obtains a near-optimal solution.

4.1. Lyapunov-Based Problem Decomposition

Within time slot t, the state of the MEC consists of the data queues $\boldsymbol{Q}(t)$ and energy queues $\boldsymbol{E}(t)$ of the WDs, expressed as $\boldsymbol{U}(t) = (\boldsymbol{Q}(t), \boldsymbol{E}(t))$. Then, we define the Lyapunov equation $L(t)$, which consists of the square sum of the data queue length and the remaining battery capacity:

$$L(t) = \frac{1}{2} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))^2 + \frac{1}{2} \sum_{m \in \mathcal{M}} (-\hat{E}_m(t))^2, \tag{27}$$

where $\hat{E}_m(t) = \Omega - E_m(t)$ represents the remaining battery capacity of WD m. $L(t)$ is a scalar measure of the length of the data queue and the size of the remaining battery capacity. It is known from Equation (27), the smaller the value of $L(t)$, the smaller the length of the data queues, which also indicates that the remaining batteries capacity of the WDs is low, and vice versa. In addition, we define the Lyapunov drift $\Delta L(t)$, which represents the expectation of the deviation of the Lyapunov equations between time slot $t+1$ and time slot t when given the network state $\boldsymbol{U}(t)$, i.e.,

$$\Delta L(t) = \mathbb{E}[L(t+1) - L(t) | \boldsymbol{U}(t)]. \tag{28}$$

In order to achieve the goal of minimizing the energy consumption of the WDs, we integrate the energy consumption function into the Lyapunov drift $\Delta L(t)$ to obtain the drift plus energy consumption function $\Delta_V L(t)$:

$$\Delta_V L(t) = \Delta L(t) + V \cdot \mathbb{E}[e(t) | \boldsymbol{U}(t)], \tag{29}$$

where V is a non-negative weight, and it represents the proportion of energy consumption $e(t)$ in $\Delta_V L(t)$. The higher the value of V, the higher the proportion of $e(t)$ in $\Delta_V L(t)$, and vice versa. By minimizing $\Delta_V L(t)$, the purpose of stabilizing queue length and minimizing energy consumption can be achieved jointly. However, as the value of V increases, the length of data queues and energy queues also increases. In other words, WDs require larger data storage devices and batteries to ensure the running of the MEC. By adjusting the value of V, we can achieve a trade-off between queue length and energy consumption in the MEC system.

Since the function $\Delta_V L(t)$ is a quadratic function of the variables in $\boldsymbol{\xi}(t)$, which are transmission power $\boldsymbol{p}(t)$, transmission time $\boldsymbol{\tau}(t)$ and amount of energy harvested $\boldsymbol{e}(t)$. Thus, it is very difficult to minimize $\Delta_V L(t)$. For easier optimization of these variables, Lemma 1 gives the upper limitation of $\Delta_V L(t)$ in time slot t. The upper limitation is a linear function composed of the variables to be optimized, thus it can greatly reduce the difficulty of the solution, and indirectly minimize $\Delta_V L(t)$.

Lemma 1. *For any feasible $\boldsymbol{\xi}(t)$, the upper bound of $\Delta L(t)$ is showed as:*

$$\Delta L(t) \leq C^{max} - \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} Q_m^n(t) \mathbb{E}\{D_m^n(t) | \boldsymbol{U}(t)\} + \sum_{m \in \mathcal{M}} \hat{E}_m(t) \mathbb{E}\{e_{m,hw}(t) | \boldsymbol{U}(t)\}, \tag{30}$$

where

$$C^{max} = \frac{1}{2} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (e_{max})^2 + (\Omega)^2 + (D_m^{n,max})^2 + (I_m^{n,max})^2) + \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} Q_m^n(t) I_m^n(t). \tag{31}$$

Proof of Lemma 1. The proof is provided in Appendix A. □

Then, we can know the upper bound of $\Delta_V L(t)$ that is following as:

$$\Delta_V L(t) \leq C^{max} + \mathbb{E}[Ve(t) - \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} Q_m^n(t) D_m^n(t) + \sum_{m \in \mathcal{M}} \hat{E}_m(t) e_{m,hw}(t) | \boldsymbol{U}(t)]. \tag{32}$$

In order to minimize the total energy consumption of WDs in the MEC system and ensure the stability of the queue at the same time, we minimize the upper bound of $\Delta_V L(t)$. In addition, since we consider the queue state $\boldsymbol{U}(t)$ within a single time slot t, we can remove expectations. In other words, we convert $\boldsymbol{P1}$ to $\boldsymbol{P2}$ based on Lemma 1 as follows:

$$\begin{aligned}&\boldsymbol{P2}: \min_{\xi(t)} Ve(t) - \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)D_m^n(t) + \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t)\\ &\quad s.t.(2)-(6),(12),(15),(17),(18),(22),(23).\end{aligned} \tag{33}$$

Problem ***P2*** is consisting of two parts linearly. Therefore, we minimize the problem into two sub-problems which are transmission optimization and battery management. In the sub-problem of transmission optimization, we optimize the transmission power $\boldsymbol{p}(t)$ and transmission time $\boldsymbol{\tau}(t)$ during tasks offloaded, while we optimize the amount of harvested energy $\boldsymbol{e}(t)$ in the sub-problem of energy management. After solving these two sub-problems, WDs update their own data queues and energy queues to prepare for the optimization of the next time slot.

We will study the solution of these two sub-problems in the following sections.

4.1.1. Transmission Optimization Problem

Considering the first term and the second term in problem ***P2***, the following transmission optimization problem can be designed to solve the transmission power $\boldsymbol{p}(t)$ and the transmission time $\boldsymbol{\tau}(t)$,

$$\begin{aligned}&\max_{\boldsymbol{p}(t),\boldsymbol{\tau}(t)} \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)D_m^n(t) - Ve(t)\\ &\quad s.t.(2)\text{–}(6),(12),(17),(18),(22).\end{aligned} \tag{34}$$

However, the maximization problem (34) is not a convex optimization problem because of the constraint (20). In order to make problem (34) easier to handle, we introduce a set of auxiliary variables $\boldsymbol{e^{var}}(t)$, i.e., $e_m^{n,var}(t) = p_m(t)\tau_m^n(t)$. Therefore, we convert problem (34) to problem (35), which is displayed as follows:

$$\begin{aligned}&\max_{\boldsymbol{e^{var}}(t),\boldsymbol{\tau}(t)} \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\{Q_m^n(t)T\frac{f_m^n(t)}{L_m(t)} + Q_m^n(t)B_{m,ul}\tau_m^n(t)\log_2(1+\frac{e_m^{n,var}(t)h_{m,ul}(t)}{\tau_m^n(t)w_0})\\ &\quad - V(e_m^{n,var}(t)+\kappa[f_m^n(t)]^3T)\}\\ &\quad s.t.\begin{cases}(2)\text{–}(6),(12),(17),(18)\\ \sum\limits_{n\in\mathcal{N}}\{e_m^{n,var}(t)+\kappa[f_m^n(t)]^3T\}\le E_m(t).\end{cases}\end{aligned} \tag{35}$$

Lemma 2. *Problem (35) is a convex optimization problem.*

Proof of Lemma 2. The proof is provided in Appendix B. □

Due to problem (35) being a convex optimization problem, it can be solved by a standard convex optimization technique, such as the interior point method. In our paper, we use the interior point method to solve the objection function in problem (35), and the computational complexity is roughly proportional to $\mathcal{O}(max((MN+M+1),F))$, where $(MN+M+1)$ is the total size of set $\boldsymbol{e^{var}}(t)$ and $\boldsymbol{\tau}(t)$, while F denotes the cost of evaluating the objection function, first and second derivatives.

4.1.2. Battery Management Problem

Considering the third term in problem ***P2***, a battery management problem can be constructed to solve $e_{m,w}(t)$, which represents the wireless energy amount, which is harvested by WD m at time slot t, respectively:

$$\begin{aligned}&\min_{\boldsymbol{e}(t)} \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t),\\ &\quad s.t.(15),(17),(18),(23).\end{aligned} \tag{36}$$

The battery management problem is a linear programming problem. In order to achieve the equation $e_{m,hw}(t) = \Omega - E_m(t)$ as much as possible, let $e^*_{m,w}(t)$ be the optimal solution to the battery management problem and the corresponding energy transfer time is defined as $l_m{}^*(t)$. If the battery of WD m can accommodate more energy $E_m(t) < \Omega$ at the beginning of time slot t, that is to say $\hat{E}_m(t) > 0$, then we need to get the wireless energy $e_{m,w}(t) = max\{\Omega - E_m(t) - \gamma_m(t), 0\}$ from the AP; otherwise, the AP will not transfer energy to WD m, meaning $e_{m,hw}(t) = 0$.

When $E_m(t) < \Omega$, we can know $\ell_m(t) \leq T - \sum\limits_{n \in \mathcal{N}} \tau^n_m(t)$ from Equation (17) and make $\ell_m(t) = T - \sum\limits_{n \in \mathcal{N}} \tau^n_m(t)$. Thus, $\ell_m{}^*(t) = min\{\ell_m, \frac{e_{m,w}(t)}{\mu p_{ap}(t) h_{m,dl}(t)}\}$ and $e^*_{m,w}(t) = \mu p_{ap}(t) h_{m,dl}(t) \ell_m{}^*(t)$. In short, by solving the battery management problem, the WD will get as much energy as possible to fill the battery.

4.2. Energy Consumption Minimization Resource Management and Allocation Algorithm (ECM-RMA)

Based on the analysis of the above two parts, we propose the ECM-RMA algorithm. In addition, the process description is given in Algorithm 1. The MEC system executes the ECM-RMA algorithm to obtain the optimal wireless transmission power $\boldsymbol{p}^*(t)$, transmission time $\boldsymbol{\tau}^*(t)$, and amount of wireless harvested energy $\boldsymbol{e}^*(t)$ at each time slot, respectively. Then, the length of data queue $\boldsymbol{Q}(t)$ and energy queue $\boldsymbol{E}(t)$ are updated according to the value of their respective variable.

Because the battery management problem has a closed-form solution, the complexity is negligible. In that way, transmission optimization problem determines the complexity of the ECM-RMA algorithm. Hence, the time complexity of the ECM-RMA algorithm is $\mathcal{O}(max((MN + M + 1), F))$.

Algorithm 1 Proposed ECM-RMA algorithm

```
Input: $\boldsymbol{Q}(t), \boldsymbol{E}(t), B_{m,ul}, h_{m,ul}(t), w_0, f^n_m(t), L_m(t), T, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$
Output: $\boldsymbol{p}^*(t), \boldsymbol{\tau}^*(t), \boldsymbol{e}^*(t), \boldsymbol{Q}(t+1), \boldsymbol{E}(t+1)$
/*Transmission Power and Transmission Time Allocation */
for $m \in \mathcal{M}$ do
    for $n \in \mathcal{N}$ do
        Solve the standard convex optimization problem in (35) to acquire the optimal power and time allocation policy;
    end for
end for
/* Battery Management */
for $m \in \mathcal{M}$ do
    if $E_m(t) < \Omega$ then
        $e_{m,w}(t) = max\{\Omega - E_m(t) - \gamma_m(t), 0\}$;
        $\ell_m(t) = T - \sum\limits_{n \in \mathcal{N}} \tau^n_m(t)$;
        $\ell_m{}^*(t) = min\{\ell_m, \frac{e_{m,w}(t)}{\mu p_{ap}(t) h_{m,dl}(t)}\}$;
        $e^*_{m,w}(t) = \mu p_{ap}(t) h_{m,dl}(t) \ell_m{}^*(t)$;
    else
        $e^*_{m,w}(t) = 0$
    end if
end for
/* Update the Queue Length */
for $m \in \mathcal{M}$ do
    Compute $E_m(t+1)$ based on Equation (11);
    for $n \in \mathcal{N}$ do
        Compute $Q^n_m(t+1)$ based on Equation (14);
    end for
end for
```

4.3. Performance Analysis

We analyze the performance of the ECM-RMA algorithm in this section. As mentioned in the previous parts, the upper limitation of $\Delta_V L(t)$ is indirectly minimizing $\Delta_V L(t)$. The proposed ECM-RMA algorithm is used to solve the problem ***P*2** in (31), which is not equal to original problem ***P*1**. Thus, how is the performance of the proposed ECM-RMA algorithm? We first show the performance of the proposed ECM-RMA algorithm in Theorem 1. Then, we reveal the energy consumption and delay trade-off achieved by the proposed ECM-RMA algorithm in Theorem 2.

4.3.1. The Performance of the Proposed ECM-RMA Algorithm

Lemma 3. *We assume that problem* **P1** *is reasonable, i.e., given average arrival data rate λ_m^n, there is at least one transmission power and time allocation scheme that satisfies all the limitations in* **P1**. *For a given average data arrival rate $\lambda_m^n + \varepsilon$, where ε is a positive number and* **P1** *is also feasible. Then, for any $\varsigma > 0$, there is at least one power and time allocation scheme $\Theta^{alg.1} = (p_m(t), \tau_m^n(t))$ that satisfies the following restrictions:*

$$\mathbb{E}(e^{alg.1}(t)|\boldsymbol{U}(t)) = \bar{e}^{alg.1} \le \bar{e}^{opt} + \varsigma, \tag{37}$$

$$\mathbb{E}(D^{alg.1}(t)|\boldsymbol{U}(t)) = \mathbb{E}(\bar{D}^{alg.1}(t)) \ge \lambda_m^n + \varepsilon, \tag{38}$$

where $\bar{e}^{alg.1}$ and $\bar{e}^{opt}$ represent the average time energy consumption under ECM-RMA algorithm (Algorithm 1) and the optimal value for solving the original problem **P1**, *respectively. $e^{alg.1}(t)$ and $D^{alg.1}(t)$ represent the energy consumption and the amount of data processed in time slot t under the scheme $\Theta^{alg.1}$, respectively.*

Proof of Lemma 3. For brevity, the proof process is omitted and similar proof in [32–34]. □

Assumption 1. *Next, we assume that there is an upper and lower bound on the average time energy consumption by ECM-RMA algorithm, i.e., $\bar{e}^{min} \le \bar{e}^{alg.1} \le \bar{e}^{max}$. Similar assumptions can be found in [32–34].*

Using Lemma 3, we can get the gap between $\bar{e}^{alg.1}$ and $\bar{e}^{opt}$, as shown in the following Theorem 1.

Theorem 1. *When* **P1** *is reasonable and $V > 0$, the boundary of $\bar{e}^{alg.1}$ is as follows:*

$$\bar{e}^{opt} \le \bar{e}^{alg.1} \le \bar{e}^{opt} + \frac{c^{max}}{V}, \tag{39}$$

where $c^{max} = \frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}((e_{max})^2 + (\Omega)^2 + (D_m^{n,max}(t))^2 + (I_m^{n,max}(t))^2)$ derived from the Proof of Lemma 1 in Appendix A.

Proof of Theorem 1. The proof is provided in Appendix C. □

Theorem 1 suggests the upper and lower bounds of the average time energy consumption $\bar{e}^{alg.1}$ is obtained by the proposed ECM-RMA algorithm. Specifically, the time average energy consumption $\bar{e}^{alg.1}$ decreases at the speed of $\mathcal{O}(1/V)$. Thus, by setting a sufficiently large V value, the average time energy consumption $\bar{e}^{alg.1}$ obtained by the ECM-RMA algorithm can be arbitrarily close to the optimal value of the original problem ***P*1**.

4.3.2. The Trade-Off between Energy Consumption and Delay

In order to reveal the energy consumption and delay trade-off achieved by the proposed ECM-RMA algorithm, we will describe the performance based on the time average data queue length, which is given in the following Theorem 2.

Theorem 2. *We assume that problem* **P1** *is reasonable, and then, in the considered MEC system, the ECM-RMA algorithm ensures that the all data queues are stable. The upper bound of the average time data queue length is shown as follows:*

$$\bar{Q} = \lim_{S \to +\infty} \frac{1}{S} \sum_{t=0}^{S-1} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} Q_m^n(t) \leq \frac{c^{max} + V(\bar{e}^{opt} - \bar{e}^{min})}{\varepsilon}, \tag{40}$$

where $\bar{e}^{min}$ is the lower bound of $\bar{e}^{opt}$, the assumption made in Section 4.3.1.

Proof of Theorem 2. The proof is provided in Appendix D. □

The average time power consumption $\bar{e}^{alg.1}$ drops at a rate of $\mathcal{O}(1/V)$ seen in Theorem 1, and, as seen from Theorem 2, the average data queue length increases at a rate of $\mathcal{O}(V)$. In other words, there is a trade-off between the average time energy consumption and the average time data queue length. In detail, with a large V value, there is less time average energy consumption but a larger average time data queue length. In addition, the trade-off can be described quantitatively as $[\mathcal{O}(1/V), \mathcal{O}(V)]$. In addition, $\bar{T}$ is denoting the average time queue delay, which is proportional to the average time data queue length at a given data arrival rate, i.e., $\bar{T} = \lim_{S \to +\infty} \frac{1}{S} \sum_{t=0}^{S-1} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \frac{Q_m^n(t)}{\lambda_m^n}$. Hence, the proposed ECM-RMA algorithm achieves an energy consumption-delay trade-off as $[\mathcal{O}(1/V), \mathcal{O}(V)]$ too. Obviously, the system resources can be effectively used, and the service quality of the user can be guaranteed by controlling the system parameter V.

5. Simulation Results

In this section, we use Matlab software (R2016a, MathWorks, Natick, Massachusetts, MA, USA) for simulation and the results of simulation are provided to evaluate the performance of the proposed ECM-RMA algorithm. We verify the reasonability and high efficiency of the ECM-RMA algorithm in minimizing energy consumption.

5.1. Simulation Settings in All Simulations, the Energy Transmitter at the AP with $p_{ap} = 3$ W

We consider a Small-scale Rayleigh fading channel model, and the channel gain $h_i = \bar{h}_i \alpha$. Here, $\bar{h}_i$ indicates that the average channel gain is determined by the geographical position of WD m and α is a random variable of a unit mean independent exponential. Specifically, $\bar{h}_i$ follows the model below:

$$\bar{h}_{m,dl} = A_d \left(\frac{3 \cdot 10^8}{4\pi f_c d_m}\right)^{d_e}, \forall m \in M. \tag{41}$$

This is a free-space path loss model, where $A_d = 4.11$ indicates the antenna gain, $f_c = 915$ MHz indicates the carrier frequency. The unit of d_e is meters, which means the distance between WD m and the AP. In addition, $d_e \geq 2$ indicates the path loss exponent. Here, we set $d_e = 2.8$. For ease of explanation, we consider $h_{m,ul} = h_{m,dl}$, $d_m = 2.5 + 0.3(m-1)$ meters and static channel model with $\alpha = 1$. In this way, $h_{m,ul} = h_{m,dl} = \bar{h}_{m,dl}$ and channel gain decreases with the increasing m.

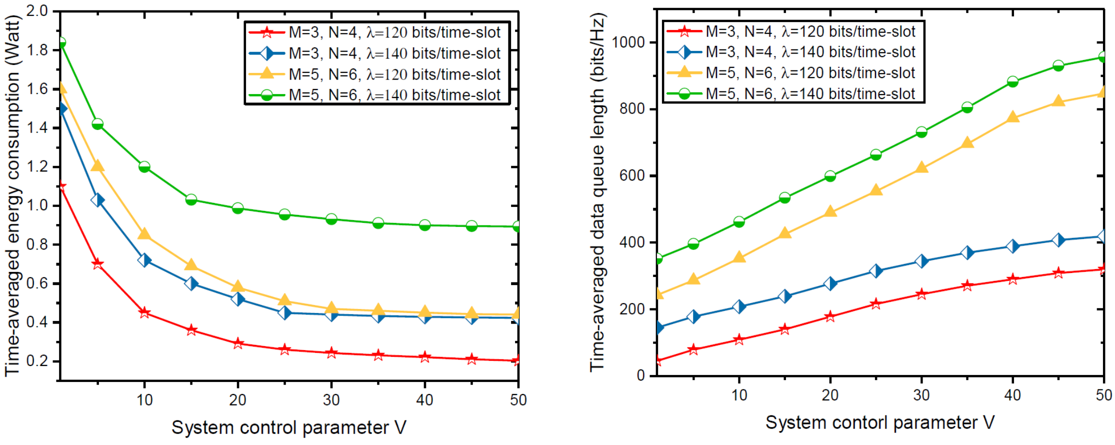
As for some parameters of the energy harvesting from the environment, we set the maximum green energy supply $\gamma_{max} = 2$, and the green energy supply γ_m is uniformly distributed in $[0, \gamma_{max}]$. Each WD has the same parameters, $\Omega = 10$ J, $k = 10^{-18}$, $f_m^n = 2.4$ GHz and $L_m = 788$ cycles/bit. In addition, each WD has the same energy queue size $R^E = \Omega = 10$ J, and each WD allocates the same amount of data storage space $R^D = 1000$ bits/Hz for each task. $R^D = 1000$ bits/Hz is the available data queue length of each task. Moreover, the completion deadline required by each WD is the same, $J_m = 0.78$ s. In addition, each task requires the same computing resources, $g_m^n = 0.5$ GHz. As for the computation on the MEC server, the computation capability of the MEC server is $f_{ap} = 4$ GHz/s [35]. During the communication process, $B_{m,ul} = 1$ MHz, $w_0 = 10^{-10}$ W, $p_{m,min} = 0.1$ Wh and $p_{m,max} = 0.2$ Wh.

The length of each time slot $T = 1$ s, and the total time length of all simulations $S =$10,000 s. At the beginning of all simulations, we set $Q_m^n(0) = 0$ and $E_m(0) = R^E, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$. Without loss of generality, the data arrival for all tasks are set to a random process with the equal average data arrival rates, i.e., $\lambda_m^n = \lambda$. Unless otherwise stated, $\lambda = 120$ bits/time-slot, $K = 2$, $\mu = 0.5$.

To evaluate the efficiency of the proposed ECM-RMA algorithm, we compare the ECM-RMA algorithm with the other two strategies, which are Baseline 1 and Baseline 2. Baseline 1 assigns equal time to tasks, and only optimizes transmission power allocation of tasks. As for Baseline 2, it only considers that WDs harvest energy from the environment. Unless otherwise stated, the other optimizations of Baseline 1 and Baseline 2 are consistent with the ECM-RMA algorithm.

5.2. Performance Achieved by the ECM-RMA Algorithm

Figure 3a shows the time-averaged energy consumption versus the system control parameter V ranging from 1 to 50. It can be seen from Figure 3 that the time-averaged energy consumption decreases with the speed of $\mathcal{O}(1/V)$ as V increases, which confirms the viewpoint in Theorem 1. In particular, when V is large enough, the time-averaged energy consumption eventually converges to the optimal value of problem $P1$, which indicates that the proposed ECM-RMA algorithm is asymptotically optimal for solving problem $P1$. In addition, we can observe that, as the number of WDs M and the number of tasks N increase, the time-averaged energy consumption will also increase. This is due to the fact that, during a certain length of time, a larger number of WDs M and a larger number of tasks N require a larger transmission rate to ensure the stability of the task data queue, which results in much more time-averaged energy consumption. In addition, part of the time-averaged energy consumption (10) can explain that the above analysis is reasonable.



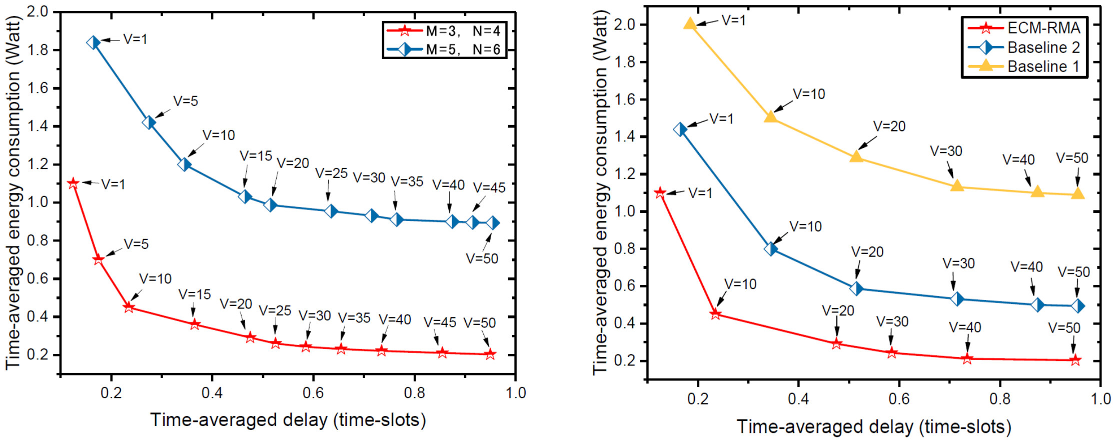
(**a**)Time-averaged energy consumption. (**b**)Time-averaged data queue length.

Figure 3. Time-averaged energy consumption and data queue length versus system control parameter V.

Figure 3b shows the time-averaged task data queue length versus the system control parameter V ranging from 1 to 50. Expectedly, the time-averaged data queue length increases with the speed of $\mathcal{O}(V)$ as V increases, which confirms the viewpoint in Theorem 2. Furthermore, we can observe that the time-averaged task data queue increases as the number of WDs M, the number of tasks N and the average data arrival rate λ increase. This is because more WDs, tasks and larger average data arrival rate λ mean more data that need to be processed, which results in an increase in the length of the data queue.

Figure 4a shows the energy consumption-delay trade-off for the WDs in the MEC system. Obviously, the larger the time-averaged delay, the smaller the time-averaged energy consumption. This is because a larger time-averaged delay means a lower quality of service (QoS), which requires only a smaller transmission rate $\nu_m(t)$. According to (1) and (10), only a small transmission power is required, which results in lesser energy consumption. Based on the analysis in Figures 3–5, there is a trade-off between energy consumption and delay, and it is quantized as $[\mathcal{O}(1/V), \mathcal{O}(V)]$. By adjusting the value of V, we can effectively control the performance of energy consumption-delay in the MEC system. More specifically, if the devices in the system have a high requirement for low latency, then a smaller V value needs to be set. Conversely, if the devices have a high requirement for low energy consumption, a large V value needs to be set.

Figure 4b shows the energy consumption-delay trade-off achieved by ECM-RMA, Baseline 1 and Baseline 2. As expected, the proposed ECM-RMA algorithm is better than Baseline 1 and Baseline 2. In more detail, in the case of the same delay, the proposed ECM-RMA algorithm consumes less energy than Baseline 1 and Baseline 2, and this also means that the ECM-RMA algorithm has the lowest latency when the same energy is consumed. This is because, compared to Baseline 1, the ECM-RMA algorithm can allocate transmission time according to the dynamics of channel and the length of date queue. Under Baseline 2, the energy harvested by WDs is unstable, and WDs cannot compute tasks, instantly resulting in long delays when the same energy is consumed.



(**a**)Energy consumption-delay trade-off achieved by ECM-RMA over different number of WDs and different number of tasks.

(**b**)Energy consumption-delay trade-off achieved by ECM-RMA, Baseline 1 and Baseline 2. The default setting is $M = 3, N = 4$.

Figure 4. Energy consumption-delay trade-off.

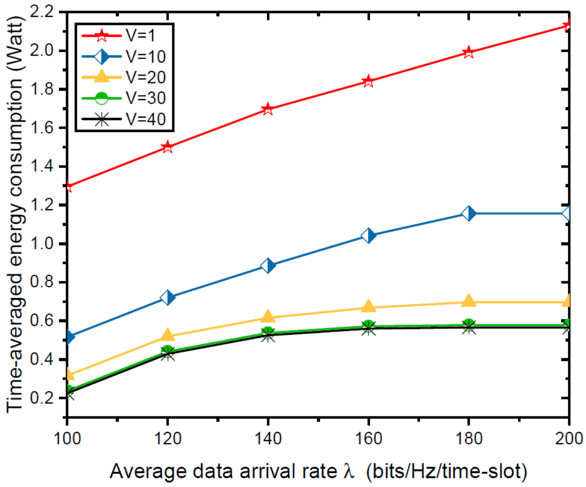


Figure 5. Time-averaged energy consumption versus average data arrival rate λ.

5.3. Impact of λ, K, μ

Now, we evaluate the impact of system variables λ, K, μ on the time-averaged energy consumption. Unless otherwise stated, $M = 3$, $N = 4$. Figure 5 shows the time-averaged energy consumption versus the average data arrival rate λ ranging from 100 to 200 under different system control parameter V. The time-averaged energy consumption increases with increasing λ. Meanwhile, the rate of increase in the time-averaged energy consumption slows down when there is a higher average data arrival rate λ due to the limited data queue size and the limited available battery capacity.

Figure 6 shows the time-averaged energy consumption versus the number of available sub-channels K under different system control parameter V. Since more available sub-channels can support more WDs for data transmission, the time-averaged energy consumption increases with K when $K \leq M$, i.e., the number of available sub-channels does not exceed the number of WDs in the MEC system.

Figure 7 shows the time-averaged energy consumption versus the energy harvesting efficiency μ under different system control parameter V. The time-averaged energy consumption monotonically increases with the increase of energy harvesting efficiency μ. This is because the wireless harvested energy of WDs is a linear function positively correlated with the energy harvesting efficiency μ according to (16), and, as μ increases, more energy can be used for wireless data transmission. However, the growth rate of the time-average energy consumption slows down with higher μ. This indicates that the time-averaged energy consumption is also affected by the battery capacity and the number of available sub-channels in case of sufficient energy supply.

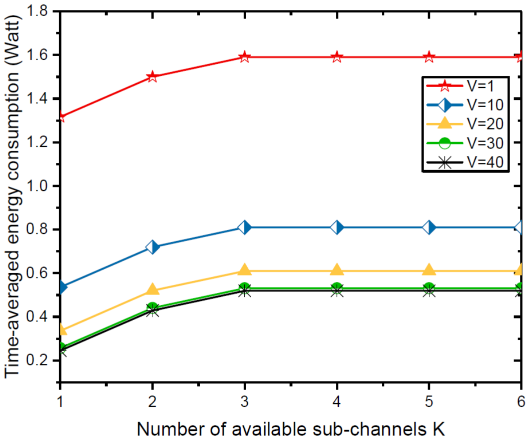


Figure 6. Time-averaged energy consumption versus number of available sub-channels K.

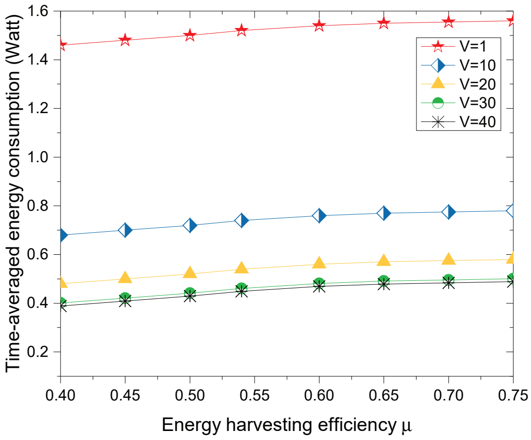


Figure 7. Time-averaged energy consumption versus energy harvesting efficiency μ.

6. Conclusions

In this paper, we investigated the time-averaged energy consumption minimization problem in a multi-user multi-task MEC system with hybrid energy harvest IoT devices. A time-average energy consumption minimization problem while satisfying data queue stability and harvesting energy availability constraints was formulated as an online stochastic optimization problem. We propose the ECM-RMA algorithm based on Lyapunov optimization technology and turn the stochastic optimization problem into two deterministic sub-problems that can be easily solved by a convex optimization technique and linear programming technique. Performance analysis shows the asymptotic optimization of the proposed ECM-RMA algorithm and energy consumption and delay trade-off. The simulation results verify the theoretical analysis and the effectiveness of the proposed ECM-RMA algorithm in minimizing the time-averaged energy consumption.

Author Contributions: H.Z., Z.C. and Y.D. conceived the idea of the paper. H.Z., Z.C., Y.D. and J.W. designed and performed the simulations; H.Z. and Y.X. analyzed the data; Z.C. contributed reagents/materials/analysis tools; H.Z. wrote the original draft; K.L. and M.L. reviewed the paper; J.W. reviewed the revised papers.

Funding: This work was supported in part by [The Major Program of National Natural Science Foundation of China] grant number [No.71633006]; [The National Natural Science Foundation of China] grant number [No. 616725407]; [China Postdoctoral Science Foundation funded project] grant number [2017M612586]; [The Postdoctoral Science Foundation of Central South University] grant number [185684]; [The Fundamental Research Funds for the Central Universities of Central South University] grant number [No. 2018zzts615].

Acknowledgments: This work was supported partially by "Mobile Health" Ministry of Education–China Mobile Joint Laboratory.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MEC	mobile edge computing
WDs	wireless devices
IoT	Internet of Things
ECM-RMA	energy-efficient joint resource management and allocation
AP	access point
EH	energy harvesting
RF	radio frequency
WPT	wireless power transfer
MDP	Markov decision processes
eDors	energy-efficient dynamic offloading resource scheduling
i.i.d	independent and identically distributed
QoS	quality of service

Appendix A. Proof of Lemma 1

Because $(max[a-b,0]+c)^2 \leq a^2+b^2+c^2-2a(b-c)$, we square both sides of (15) and (21) to get (A1) and (A2), respectively. In addition, (A1) and (A2) are shown below:

$$Q^n_m(t+1))^2 \leq (Q^n_m(t))^2 + (D^n_m(t))^2 + (I^n_m(t))^2 - 2Q^n_m(t)(D^n_m(t) - I^n_m(t)), \tag{A1}$$

$$(\hat{E}_m(t+1))^2 \leq (\hat{E}_m(t))^2 + (\sum_{n\in\mathcal{N}} e^n_{m,total}(t))^2 + (e_{m,hw}(t))^2 - 2\hat{E}_m(t)(\sum_{n\in\mathcal{N}} e^n_{m,total}(t) - e_{m,hw}(t)). \tag{A2}$$

Summing all WDs $m \in \{1, 2, ...M\}$ for (A1) and (A2), we can get (A3) and (A4), respectively. In addition, (A3) and (A4) are shown below:

$$\begin{aligned}&\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\frac{1}{2}((Q_m^n(t+1))^2-(Q_m^n(t))^2)\le\\&\frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}((D_m^n(t))^2+(I_m^n(t))^2)-\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}Q_m^n(t)(D_m^n(t)-I_m^n(t)),\end{aligned} \tag{A3}$$

$$\begin{aligned}&\sum_{m\in\mathcal{M}}\frac{1}{2}((\hat{E}_m(t+1))^2-\hat{E}_m(t))^2)\le\\&\frac{1}{2}\sum_{m\in\mathcal{M}}((\sum_{n\in\mathcal{N}}e_{m,total}^n(t))^2+(e_{m,hw}(t))^2)-\sum_{m\in\mathcal{M}}\hat{E}_m(t)(\sum_{n\in\mathcal{N}}e_{m,total}^n(t)-e_{m,hw}(t)).\end{aligned} \tag{A4}$$

Therefore, the Lyapunov drift $\Delta L(t)$ can be given by

$$\begin{aligned}&\Delta L(t)\le\\&\frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\mathbb{E}\{((D_m^n(t))^2+(I_m^n(t))^2)|\boldsymbol{U}(t)\}-\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}Q_m^n(t)\mathbb{E}\{(D_m^n(t)-I_m^n(t))|\boldsymbol{U}(t)\}\\&+\frac{1}{2}\sum_{m\in\mathcal{M}}\mathbb{E}\{((\sum_{n\in\mathcal{N}}e_{m,total}^n(t))^2+(e_{m,hw}(t))^2)|\boldsymbol{U}(t)\}-\sum_{m\in\mathcal{M}}\hat{E}_m(t)\mathbb{E}\{(\sum_{n\in\mathcal{N}}e_{m,total}^n(t)-e_{m,hw}(t))|\boldsymbol{U}(t)\}.\end{aligned} \tag{A5}$$

During a time slot, a fixed amount of data can be processed in a WD according to Equation (7). In addition, the amount of data to be transmitted also has upper and lower bounds. This is because the transmission power has upper and lower limits under the determined transmission time. Therefore, we know $D_m^{n,min} \le D_m^n(t) \le D_m^{n,max}$. In addition, $\sum_{n\in\mathcal{N}} e_{m,total}^n(t)$ cannot exceed the length of its energy queue at time slot t, so we have the upper bound of energy consumption, e.g., e_{max}. As for $e_{m,hw}(t)$, it cannot exceed the size of battery capacity, so we have the upper bound of amount of harvested energy, e.g., $e_{m,hw}(t) \le \Omega$. We use the upper bound $D_m^n(t) \le D_m^{n,max}, I_m^n(t) \le I_m^{n,max}, e_{m,total}^n(t) \le e_{max}, e_{m,hw}(t) \le \gamma_{max}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$ and define $c^{max} = \frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}((e_{max})^2 + (\Omega)^2 + (D_m^{n,max}(t))^2 + (I_m^{n,max}(t))^2)$. In addition, $I_m^n(t)$ is the amount of data arrive to WD m at the end of time slot t, so both the data queue length $Q_m^n(t)$ and the corresponding data queue state $\boldsymbol{U}(t)$ are independent of $I_m^n(t)$ at the current time slot t. As a result, $I_m^n(t)$ is independent of the current power allocation $p_m(t)$ and time allocation $\tau_m^n(t)$. In other words, $\mathbb{E}\{I_m^n(t)|\boldsymbol{U}(t)\}$ is a constant. We can define C_{max} as follows:

$$C^{max} = c^{max} + \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t) I_m^n(t). \tag{A6}$$

Therefore, we have

$$\Delta L(t) \le C^{max} - \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)\mathbb{E}\{D_m^n(t)|\boldsymbol{U}(t)\} + \sum_{m\in\mathcal{M}} \hat{E}_m(t)\mathbb{E}\{e_{m,hw}(t)|\boldsymbol{U}(t)\}. \tag{A7}$$

Lemma 1 is eventually proved.

Appendix B. Proof of Lemma 2

First of all, we will show that the binary function $f(x_1, x_2) = -x_1 \ln(1 + \frac{x_2}{x_1})$ is convex on x_1 and x_2. The reason is as follows:

The Hessian matrix of $f(x_1, x_2) = -x_1 \ln(1 + \frac{x_2}{x_1})$ is

$$H = \begin{bmatrix} \frac{x_2^2}{(x_1+x_2)^2 x_1} & -\frac{x_2}{(x_1+x_2)^2} \\ -\frac{x_2}{(x_1+x_2)^2} & \frac{x_1}{(x_1+x_2)^2} \end{bmatrix}. \tag{A8}$$

The eigenvalues of H are $k_1 = 0$ and $k_2 = \frac{x_1^2+x_2^2}{x_1^3+2x_1^2x_2+x_1x_2^2}$, respectively, and H is positive semidefininite. Therefore, $f(x_1, x_2)$ is convex on x_1 and x_2.

For (35), let $x_1 = \tau_m^n(t)$, $x_2 = \frac{e_m^{n,var}(t)h_{m,ul}(t)}{w_0}$. According to (A8), since $f(x_1, x_2)$ is convex, $\eta_m^n(\tau_m^n(t), e_m^{n,var}(t)) = -\frac{Q_m^n(t)B_{m,ul}f(x_1,x_2)}{\ln 2}$ is concave. In the issue, $\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\eta_m^n(\tau_m^n(t), e_m^{n,var}(t))$ is concave too. In addition, the first term in (35) is independent of the variable to be optimized, whereas the third term and fourth term are linear functions of $e_m^{n,var}(t)$. Therefore, the objection problem in (35) is concave. Clearly, the constraints in Equation (35) are linear of $\tau_m^n(t)$ and $e_m^{n,var}(t)$. Thus, the feasible region of the problem in Equation (35) is convex.

In summary, the objection problem in (35) is a convex optimization problem. Lemma 2 is eventually proved.

Appendix C. Proof of Theorem 1

Now, we will show the performance gap caused by Lyapunov optimization theory compared to the original problem $P1$. First, we use Algorithm 1 to minimize the right hand of (32), and we have

$$\Delta_V L(t) \leq C^{max} + \mathbb{E}[Ve^{alg.1}(t) - \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)D^{alg.1}(t) + \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t)|\boldsymbol{U}(t)]. \tag{A9}$$

Then, substituting (37) and (38) into (A9),

$$\begin{aligned}&\Delta_V L(t) \leq \\ &c^{max} + \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)I_m^n(t) + \mathbb{E}[V(\bar{e}^{opt} + \varsigma) - \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t)(\lambda_m^n + \varepsilon) + \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t)|\boldsymbol{U}(t)].\end{aligned} \tag{A10}$$

In addition, because arrival data rate λ_m^n represents the amount of data entering per second, $I_m^n(t) = \lambda_m^n$. Now, we can get

$$\begin{aligned}&\Delta_V L(t) \leq \\ &c^{max} + \mathbb{E}[V(\bar{e}^{opt} + \varsigma) - \varepsilon \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t) + \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t)|\boldsymbol{U}(t)].\end{aligned} \tag{A11}$$

Setting a limit as $\varsigma \rightarrow 0$,

$$\Delta_V L(t) \leq c^{max} + V\bar{e}^{opt} - \varepsilon \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t) + \sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t). \tag{A12}$$

Then, accumulating (A12) from $t = 1$ to $t = S$, we can get

$$\begin{aligned}&\mathbb{E}[L(T)] - \mathbb{E}[L(1)] + V\sum_{t\in\mathcal{T}} \mathbb{E}[e^{alg.1}(t)|\boldsymbol{U}(t)] \leq \\ &S(c^{max} + V\bar{e}^{opt}) - \varepsilon \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_m^n(t) + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}} \hat{E}_m(t)e_{m,hw}(t).\end{aligned} \tag{A13}$$

Finally, letting (A13) divide by VS and making $S \rightarrow \infty$, we can get

$$e^{alg.1}(t) \leq \bar{e}^{opt} + \frac{c^{max}}{V}. \tag{A14}$$

Theorem 1 is eventually proved.

Appendix D. Proof of Theorem 2

Since $Q_m^n(t) \geq 0, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}$, we remove the second item on the right side of (A12); then, (A12) can be simplified to

$$\Delta_V L(t) \leq c^{max} + V\bar{e}^{opt} + \sum_{m \in \mathcal{M}} \hat{E}_m(t) e_{m,hw}(t), \tag{A15}$$

$$\Delta L(t) \leq c^{max} + \sum_{m \in \mathcal{M}} \hat{E}_m(t) e_{m,hw}(t). \tag{A16}$$

We learn $C_m^n(t) \leq C_m^{n,max}(t), \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}$, and $D_m^n(t) \leq D_m^{n,max}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}$ from this paper. Thus, $\Delta L(t) \leq A$, where A is a finite positive constant. According to (28), we can get

$$\mathbb{E}[L(t+1) - L(t) | \boldsymbol{U}(t)] \leq A. \tag{A17}$$

Accumulating (A16) from $t = 1$ to $t = T$, we acquire

$$\mathbb{E}[L(T) - L(1) | \boldsymbol{U}(t)] \leq TA. \tag{A18}$$

Substituting (27) into (A18), we have

$$\frac{1}{2}\mathbb{E}[\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))^2] + \frac{1}{2}\mathbb{E}[\sum_{m \in \mathcal{M}} (-\hat{E}_m(t))^2] - \mathbb{E}[L(1)] \leq TA, \tag{A19}$$

$$\mathbb{E}[\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))^2] \leq 2TA + 2\mathbb{E}[L(1)] - \mathbb{E}[\sum_{m \in \mathcal{M}} (-\hat{E}_m(t))^2], \tag{A20}$$

$$\mathbb{E}[\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))^2] \leq 2TA + 2\mathbb{E}[L(1)]. \tag{A21}$$

Then, we convert (A21) to (A22)

$$\mathbb{E}[\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))] \leq \sqrt{2TA + 2\mathbb{E}[L(1)]}. \tag{A22}$$

Letting (A22) divide by S and making $S \to \infty$, we can get

$$\lim_{S \to +\infty} \frac{1}{S}\mathbb{E}[\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} (Q_m^n(t))] < \infty. \tag{A23}$$

From (A2), we know that the ECM-RMA algorithm can guarantee the stability of all data queues in the considered MEC system. Moreover, letting (A13) divide by εS and making $S \to \infty$, we can get

$$\bar{Q} = \lim_{S \to +\infty} \frac{1}{S} \sum_{t=0}^{S-1} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} Q_m^n(t) \leq \frac{c^{max} + V(\bar{e}^{opt} - \bar{e}^{min})}{\varepsilon}, \tag{A24}$$

where the inequality contained in (A24) follows from Assumption 1. Theorem 2 is eventually proved.

References

1. Mao, S.; Leng, S.; Yang, K.; Zhao, Q.; Liu, M. Energy Efficiency and Delay Tradeoff in Multi-User Wireless Powered Mobile-Edge Computing Systems. In Proceedings of the GLOBECOM 2017–2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
2. Bi, S.; Zhang, Y.J.A. Computation Rate Maximization for Wireless Powered Mobile-Edge Computing with Binary Computation Offloading. *IEEE Trans. Wirel. Commun.* **2017**, *17*, 4177–4190. [CrossRef]

3. Wu, J.; Chen, Z.; Zhao, M. Information cache management and data transmission algorithm in opportunistic social networks. *Wirel. Netw.* **2018**, 1–12. [CrossRef]
4. Park, D.; Kim, S.; An, Y.; Jung, J.Y. LiReD: A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks. *Sensors* **2018**, *18*, 2110. [PubMed]
5. Premsankar, G.; Francesco, M.D.; Taleb, T. Edge Computing for the Internet of Things: A Case Study. *IEEE Internet Things J.* **2018**, *5*, 1275–1284. [CrossRef]
6. Wang, F.; Xu, J.; Wang, X.; Cui, S. Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems. *IEEE Trans. Wirel. Commun.* **2017**, *17*, 1784–1797. [CrossRef]
7. Wu, J.; Chen, Z.; Zhao, M. Weight distribution and community reconstitution based on communities communications in social opportunistic networks. *Peer-to-Peer Netw. Appl.* **2018**, 1–9. [CrossRef]
8. Mahapatra, C.; Moharana, A.; Leung, V. Energy Management in Smart Cities Based on Internet of Things: Peak Demand Reduction and Energy Savings. *Sensors* **2017**, *17*, 2812. [PubMed]
9. Chen, Z.; Guo, L.; Zhang, D.; Chen, X. Energy and channel transmission management algorithm for resource harvesting body area networks. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 155014771875923. [CrossRef]
10. Jia, W.U.; Chen, Z.; Zhao, M. Effective information transmission based on socialization nodes in opportunistic networks. *Comput. Netw.* **2017**, *129*, 297–305.
11. Nan, Y.; Li, W.; Bao, W.; Delicato, F.C.; Pires, P.F.; Dou, Y.; Zomaya, A.Y. Adaptive Energy-aware Computation Offloading for Cloud of Things Systems. *IEEE Access* **2017**, *5*, 23947–23957. [CrossRef]
12. Ko, S.W.; Kim, S.L. Impact of Node Speed on Energy-Constrained Opportunistic Internet-of-Things with Wireless Power Transfer. *Sensors* **2018**, *18*, 2398. [CrossRef] [PubMed]
13. Yang, J.; Yang, Q.; Kwak, K.S.; Rao, R.R. Power–CDelay Tradeoff in Wireless Powered Communication Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3280–3292. [CrossRef]
14. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [CrossRef]
15. Kao, Y.H.; Krishnamachari, B.; Ra, M.R.; Bai, F. Hermes: Latency Optimal Task Assignment for Resource-constrained Mobile Computing. *IEEE Trans. Mob. Comput.* **2017**, *16*, 3056–3069. [CrossRef]
16. Zhang, W.; Wen, Y.; Wu, D.O. Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 81–93. [CrossRef]
17. Khalili, S.; Simeone, O. Inter-layer per-mobile optimization of cloud mobile computing: A message-passing approach. *Trans. Emerg. Telecommun. Technol.* **2016**, *27*, 814–827. [CrossRef]
18. Lorenzo, P.D.; Barbarossa, S.; Sardellitti, S. Joint Optimization of Radio Resources and Code Partitioning in Mobile Edge Computing. *arXiv* **2013**, arXiv:1307.3835.
19. Mahmoodi, S.E.; Subbalakshmi, K.P.; Sagar, V. Cloud offloading for multi-radio enabled mobile devices. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5473–5478.
20. You, C.; Huang, K.; Chae, H.; Kim, B.H. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [CrossRef]
21. Barbarossa, S.; Sardellitti, S.; Lorenzo, P.D. Joint allocation of computation and communication resources in multiuser mobile cloud computing. In Proceedings of the 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Darmstadt, Germany, 16–19 June 2013; Volume 395, pp. 26–30.
22. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [CrossRef]
23. Lyu, X.; Tian, H.; Sengul, C.; Zhang, P. Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3435–3447. [CrossRef]
24. Mao, Y.; Zhang, J.; Song, S.H.; Letaief, K.B. Power-Delay Tradeoff in Multi-User Mobile-Edge Computing Systems. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016.
25. Guo, S.; Xiao, B.; Yang, Y.; Yang, Y. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. In Proceedings of the IEEE INFOCOM 2016—The IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.

26. Yu, Y.; Zhang, J.; Letaief, K.B. Joint Subcarrier and CPU Time Allocation for Mobile Edge Computing. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
27. Zhao, T.; Zhou, S.; Guo, X.; Zhao, Y. A Cooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-Aware Mobile Cloud Computing. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
28. Chen, M.H.; Dong, M.; Liang, B. Joint offloading decision and resource allocation for mobile cloud with computing access point. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 3516–3520.
29. Xu, J.; Ren, S. Online Learning for Offloading and Autoscaling in Renewable-Powered Mobile Edge Computing. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
30. Mao, Y.; Zhang, J.; Letaief, K.B. Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3590–3605. [CrossRef]
31. Hao, Y.; Chen, M.; Hu, L.; Hossain, M.S.; Ghoniem, A. Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access* **2018**, *6*, 11365–11373. [CrossRef]
32. Li, Y.; Sheng, M.; Shi, Y.; Ma, X. Energy Efficiency and Delay Tradeoff for Time-Varying and Interference-Free Wireless Networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 5921–5931. [CrossRef]
33. Neely, M. Stochastic Network Optimization with Application to Communication and Queueing Systems. *Synth. Lect. Commun. Netw.* **2010**, *3*, 1–211. [CrossRef]
34. Li, Y.; Sheng, M.; Wang, C.X.; Wang, X.; Shi, Y.; Li, J. Throughput–Delay Tradeoff in Interference-Free Wireless Networks With Guaranteed Energy Efficiency. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 1608–1621. [CrossRef]
35. Zhang, K.; Mao, Y.; Leng, S.; Zhao, Q.; Li, L.; Peng, X.; Pan, L.; Maharjan, S.; Zhang, Y. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks. *IEEE Access* **2017**, *4*, 5896–5907. [CrossRef]