This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

Energy-Delay Tradeoff for Dynamic Offloading in Mobile-Edge Computing System with Energy Harvesting Devices

Guanglin Zhang, *Member, IEEE*, Wenqian Zhang, Yu Cao, Demin Li, and Lin Wang, *Member, IEEE*

***Abstract*—Mobile-edge Computing (MEC) has aroused significant attention for its performance to accelerate application's operation and enrich user's experience. With the increasing development of green computing, energy harvesting (EH) is considered as an available technology to capture energy from circumambient environment to supply extra energy for mobile devices. In this paper, we propose an online dynamic tasks assignment scheduling to investigate the tradeoff between energy consumption and execution delay for MEC system with EH capability. We formulate it into an average weighted sum of energy consumption and execution delay minimization problem of mobile device with the stability of buffer queues and battery level as constraints. Based on Lyapunov optimization method, we obtain the optimal scheduling about the CPU-cycle frequencies of mobile device and transmit power for data transmission. Besides, the dynamic online tasks offloading strategy is developed to modify the data backlogs of queues. The performance analysis shows the stability of the battery energy level and the tradeoff between energy consumption and execution delay. Moreover, the MEC system with EH devices and task buffers implements the high energy efficient and low latency communications. The performance of the proposed online algorithm is validated with extensive trace-driven simulations.**

***Index terms*-mobile-edge computing, energy harvesting, online dynamic offloading, energy-delay tradeoff, Lyapunov optimization.**

I. Introduction

The growing preference for mobile devices has promoted the rapid development of mobile applications [1], leading to extremely intensive data computing [2] and more rigorous requirements for mobile devices, i.e., execution and storage ability, as well as battery energy [3]. Mobile edge computing (MEC) as a new paradigm that relies on its internal provided wireless network to acquire computational capabilities [4]. Compared with the traditional remote cloud (*e.g.*, Amazon, Google, *etc.*) [5], offloading workload to MEC server not only reduce the congestion of data transmission, but also save the energy consumption of user [6], [7] on account of MEC server physically close to users [8]. Therefore, it is pressing to take advantage of the MEC features to share the intensive workload of users (through offloading) [9] and reduce traffic overhead of industrial applications [10]. In addition, it's worthwhile to leverage energy harvesting (EH) technology to capture the green energy (*e.g.*, solar, wind and/or solar radiation, *etc.*) for charging battery constantly [11]. In Conclusion, the MEC system with EH devices improves its computational service capabilities [12], [13], which is also consistent with the notion of green communication [14].

G. Zhang, W. Zhang, Y. Cao, and D. Li are with the College of Information Science and Technology, Donghua University; Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, Shanghai, China. (email: {glzhang, deminlin}@dhu.edu.cn, {zhangwenqian, yucao} @mail.dhu.edu.cn). Corresponding author: Guanglin Zhang.

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

With the integration of EH into the MEC system and the premise of guaranteeing the calculation performance of the system. It has some new challenges to realize the stability of the battery energy level in the long-term evolution. It is essential to make more efficient use of the renewable resources in the surrounding environment. Moreover, the task assignment decisions need to consider more relevant information, i.e., the size of the arrived task, the EH process, the battery energy level and the status of the channel, which is difficult to achieve for mobile devices with limited computing capability and storage capacity. We will design a new and effective method for MEC system that takes EH devices into consideration.

A. Contributions

In this paper, we investigate a problem for minimizing the average weighted summation of energy consumption and execution delay of mobile device with the stabilizing of battery level and buffer queues (offloading buffers and downloading buffers). We propose an online dynamic Lyapunov optimization-based offloading algorithm to decide the tasks assignment, e.g., the arrived tasks are distributed to local execution, or to MEC server execution, or to drop. In particular, we obtain the tradeoff between energy consumption and execution delay by adjusting the introduced parameters. Specifically, we reduce execution delay by increasing the running frequency of the local CPU, or save energy consumption by choosing a good channel for data transmission. Due to the time-dependent battery energy and buffer queues, the design becomes more complicated. Compared to the MEC system without the EH devices and data buffers, the calculated service performance of our proposed system has improved.

The major contributions are summarized following:

- We investigate the tasks assignment problem which are workload balancing scheduling (for task offloading) and downloading scheduling (for data results downloading), obtaining the energy-delay tradeoff in MEC system with EH devices and multiple applications. The tasks assignment decision relies on the various time-dependent information, including the arrived data of execution tasks,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

channel condition, the congestion of transmission, the amount of harvested renewable energy and energy level of the battery. It's challenging to take these various and practical information into consideration to obtain the long-term optimal system performance, and it's also worthwhile for the design of future MEC systems.

- We formulate an average weighted sum of energy consumption and execution delay minimization problem under the condition of considering the stability of battery and buffer backlogs. We develop the online dynamic Lyapunov optimization-based offloading algorithm (ODLOO) to implement tasks assignment while measure the tradeoff between energy consumption and execution delay. It's crucial that the proposed algorithm requires current system's information purely. Besides, the time complexity of ODLOO algorithm characterizes as $O(n)$, which is suitable for the operation of mobile device.
- We prove the correctness of the perturbation parameter of battery and show its bound, which is important for the stability of battery. Besides, we show the gap between the original problem and our proposed algorithm. It shows that we can achieve asymptotic optimization by adjusting the control parameter so that it approaches infinite. Furthermore, the optimal scheduling achieves the tradeoff between energy consumption and execution delay by adjusting the parameters. Moreover, simulation results affirms that the MEC system with EH devices improves its calculating service capabilities by comparing with three benchmark polices, i.e., CPU-Only, MEC-Only and Dynamic Offloading (Greedy).

II. System Model and Problem Formulation

We consider a general MEC system consisting of one MEC server, one mobile device with the energy harvesting components as shown in Fig. 1. The mobile device running N applications, each application has its corresponding two buffer queues (task queue and return queue). We denote the set of applications as $\mathcal{N} \triangleq \{0, 1..., N\}$. We assume that $\mathcal{N}'$ applications are executed by MEC server, among which $\mathcal{N}' \in \mathcal{N}$. The set of time slot as $\mathcal{T} \triangleq \{0, 1...\}$ and the length of each time slot is τ. We denote the collections of two buffers namely task queue $\mathcal{Q}(t)$ and return queue $\mathcal{Q}^D(t)$, respectively. The arrived tasks of applications are stored in task queues firstly to wait task assignment. The output results are downloaded by mobile device from the return queues after MEC server execution. Moreover, the data transmission between mobile device and MEC server based on the wireless channel and D2D method with high efficiency and low latency.

A. Data Transmission and Queuing Model

In each time slot, the execution tasks $A_i(t)$ (in bits) generated by the ith application are modeled as independent and identically distributed (i.i.d.). We denote the set of arrived tasks from applications as $\mathcal{A}(t) = \{A_1(t), A_2(t)..., A_N(t)\}$, the arrived tasks from application i in time slot t will store in $Q_i(t)$ to wait tasks assignment, where the $Q_i(t)$ (in bits)

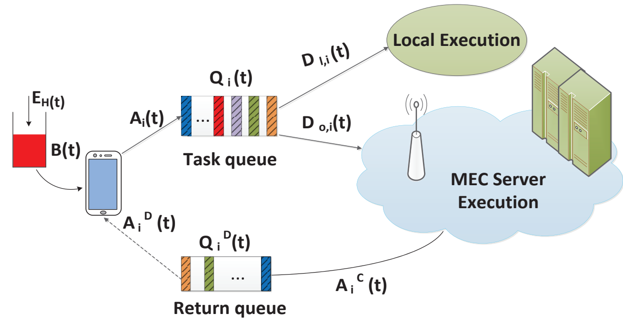


Fig. 1: The system model.

represents the length of task queue i. The set of task queue at the beginning of time slot t is denoted as

$$\mathcal{Q}(t) = \{Q_1(t), Q_2(t)..., Q_N(t)\}. \tag{1}$$

In addition, we denote the data size executed by local CPU as $D_{l,i}(t)$ (in bits), the data size computed by MEC server is $D_{o,i}(t)$ (in bits). The set of the data size for local execution and MEC execution are defined as $\mathcal{D}_L(t) = \{D_{l,1}(t), D_{l,2}(t)..., D_{l,N}(t)\}$ and $\mathcal{D}_O(t) = \{D_{o,1}(t), D_{o,2}(t)..., D_{o,N}(t)\}$, respectively. We define $Q_i^D(t)$ (in bits) as the length of return queue for storing the data results returned by MEC server of application i in time slot t, and the set of return queue at the end of the time slot t as

$$\mathcal{Q}^D(t) = \{Q_1^D(t), Q_2^D(t)..., Q_N^D(t)\}. \tag{2}$$

We define the set of arrived data results to each data return queue as $\mathcal{A}^C(t) = \{A_1^C(t), A_2^C(t)..., A_N^C(t)\}$. Especially, we assume that the input and output of MEC server are the same in numerical values. Thus, we have $|D_{o,i}(t)| = |A_i^C(t)|$, where $A_i^C(t)$ is the arrived data size to $Q_i^D(t)$ of application i at time slot t. We define the size of downloadable data results from return queue to mobile device as $A_i^D(t)$, the set of downloaded data results denote as $\mathcal{A}^D(t) = \{A_1^D(t), A_2^D(t)..., A_N^D(t)\}$. Without loss of generality, we assume that the MEC server provides the infinite computation capability compared to mobile device, the computing time at MEC server is viewed as a constant. The dynamics of the task queue backlog $Q_i(t)$ is

无限计算能力MEC

$$Q_i(t+1) = \max\{Q_i(t) + A_i(t) - D_i(t), 0\}, i \in \mathcal{N}, \tag{3}$$

where $D_i(t) = D_{l,i}(t) + D_{o,i}(t) + D_{d,i}(t)$. Note that $D_{d,i}(t)$ is the data size of dropping tasks, i.e., if the system has insufficient computation resources or long time waiting for the tasks $A_i(t)$ arrived at time slot t, some tasks in the task queue will be dropped. We assume that $|D_{d,i}(t)| = |A_i(t)|$. In other words, if $|A_i(t)| > |D_{d,i}(t)|$, the length of task queue will increase and the queue will be unstable. In addition, the dynamics of the data results return queue backlog $Q_i^D(t)$ is

$$Q_i^D(t+1) = \max\{Q_i^D(t) + A_i^C(t) - A_i^D(t), 0\}, i \in \mathcal{N}', \tag{4}$$

where $|A_i^C(t)| = |D_{o,i}(t)|$.

$$\overline{Q} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\{Q_i(t) + Q_i^D(t)\}, \tag{5}$$

we adopt (5) to represent the state of buffer queues at time slot t [15], and we consider the system to be stable if the average length of $\overline{Q}$ is bounded.

B. Computing Model and Energy Consumption

The mobile device makes the tasks assignment decision at each time slot according to the proposed algorithm, the tasks will be executed by local CPU or MEC server, or be dropped. When the battery has enough energy, the tasks either compute by local CPU to reduce transmission delay, or offload to MEC server to save energy consumption of mobile device. When the power of the battery is deficient, the tasks will be dropped. According to the principle of First-In-First-Out (FIFO), the arrived tasks will be choreographed and executed. In addition, the length of one time slot τ is made up of two parts, part is the time for task processing, and the other part is data transfer time. Due to the limited computing power of the system, the time for data execution in one time slot can not exceed the task processing deadline τ_d. In order to ensure that the assigned tasks can be completed in one time slot, under the condition that the total execution time of the tasks remains unchanged, a large tasks are divided into smaller subtasks to execute, which depends on the two processing parameters of the system's processing capability and the processing deadline τ_d.

We denote three indicators to decide tasks assignment, which are $\alpha^L(t)$, $\alpha^O(t)$ and $\alpha^d(t)$, respectively. The tasks assignment decision is described as $\boldsymbol{\alpha}(t) \triangleq \{\alpha^L(t), \alpha^O(t), \alpha^d(t)\}$, and we have $\alpha^L(t) \cup \alpha^O(t) \cup \alpha^d(t) = \mathcal{N}$. When $i \in \alpha^L(t)$, the mobile device computes the tasks of application i from one of the appropriate task queue. Similarly, when $i \in \alpha^O(t)$, the tasks will be allocated to MEC server execution. Particularly, we define $\alpha^d(t)$ as the decision of dropping tasks. When $i \in \alpha^d(t)$, the arrived tasks will be dropped, the system will stop working without energy consumption. Furthermore, the decision indicators should satisfy the following operation constraint:

$$\alpha^L(t) \cap \alpha^O(t) \cap \alpha^d(t) = \emptyset, t \in \mathcal{T}, \tag{6}$$

it ensures the tasks $A_i(t)$ can not select local execution, MEC execution, or dropping in the same time slot t.

Local Execution Model: When tasks are computed by local CPU, the operation of CPU accounted for major energy consumption (regardless of the human factors, *i.e.*, the power consumption of the screen depends on users habits). In particular, the energy consumption is proportional to the square of the frequency of mobile device. For tasks of application i being executed locally at time slot t, we denote the corresponding CPU-cycle frequency as $f_i(t)$, which not exceeds its maximum value $f_{\max}$, and the power consumption is approximated as

$$P_{l,i}(t) = k \cdot f_i^2(t), t \in \mathcal{T} \tag{7}$$

where k is the effective energy coefficient related to chip architecture. We have the data size computed by local CPU

$$D_{l,i}(t) = \tau f_i(t) L_i^{-1}, i \in \alpha^L(t), t \in \mathcal{T}, \tag{8}$$

where L_i is the needed CPU-cycles for processing one bit input task of ith application. Specially, $i \in \alpha^L(t)$ means that tasks are executed by local CPU purely, we have the overall energy consumption of local execution as follow

$$P_L(t) = \sum_{i=1}^{N} P_{l,i}(t), t \in \mathcal{T}. \tag{9}$$

MEC Server Execution Model: When $i \in \alpha^O(t)$, the tasks are assigned to MEC server execution purely, the returned data results are downloaded by mobile device. Notice that the data transmission depends on the status of the wireless channel. Offloading tasks to MEC server can reduce the heavy workload of mobile device while increase the processing time delay. Therefore, we should consider the tradeoff between energy consumption and execution delay.

1) Offloading tasks to MEC server: When tasks offloaded to MEC server, the data transmission occupies major energy consumption and time delay. The data size processed by MEC server of application i at tth time slot is denoted as

$$D_{o,i}(t) = \omega\tau log_2\left(1 + \frac{H_i(t)P_{o,i}(t)}{N_0\omega}\right), i \in \alpha^O(t), \tag{10}$$

where $H_i(t)$ is the channel power gain from the mobile device to MEC server at the tth time slot, $P_{o,i}(t)$ is transmit power, ω is the bandwidth. We denote the total energy consumption for MEC server execution as $P_O(t) = \sum_{i=1}^{N} P_{o,i}(t)$.

2) Downloading data results by mobile device: When tasks of application i are executed by MEC server, the data results are stored in $Q_i^D(t)$ and downloaded by mobile device. The downloadable data results from MEC server to mobile device are expressed as

$$A_i^D(t) = arg \max_{\{\Sigma_{k=1}^{b} A_i^C(k)\}} \left\{\Sigma_{k=1}^{b} A_i^C(k) \leq w_D(t)\right\}, i \in \mathcal{N}', \tag{11}$$

it means that we obtain the maximum downloadable data size by finding a suitable task b to calculate $\Sigma_{k=1}^{b} A_i^C(k)$. Where $w_D(t)$ is the downlink wireless data rate, it depends on the condition of wireless channel only. $A_i^C(k)$ is the data results output of MEC server of task k of application i. We denote the whole energy consumption for data results downloading as $P_D(t)$, and show the definition of $P_D(t)$ in our simulations. We select a suitable wireless channel environment for data transmission in order to save energy consumption. Waiting for channel to achieve a good condition blindly, the execution delay and the length of return queues will increase. It's worthwhile to balance the energy-delay in MEC system.

C. The Energy Harvesting Model and Battery

The EH devices capture the renewable energy sources to supply the mobile device. $E_H(t)$ is the actual harvested energy to battery among different time slot with the maximum value of $E_H^{\max}$, we assume that $E_H(t)$'s are i.i.d. and we have

$$0 \leq E_H(t) \leq E_H^{\max}, t \in \mathcal{T}. \tag{12}$$

The harvested energy is stored in the battery, and provides energy for local computation and data transmission of MEC server execution from the next time slot. We denote the battery energy level at the beginning of time slot t as $B(t)$, the bound of the battery level at time slot t is given by

$$E_{\min} \leq B(t) \leq E_{\max}, t \in \mathcal{T}, \tag{13}$$

where $0 < E_{\min} \leq E_{\max}$, $E_{\max}$ and $E_{\min}$ are the maximum and minimum discharged energy of the battery, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

The overall energy consumption of the system in time slot t is denoted as $P(t)$ and expressed as

$$P(t) = P_L(t) + P_O(t) + P_D(t), t \in \mathcal{T}, \tag{14}$$

notice that we added the term $\phi \cdot \mathbf{1}(i \in \alpha^d(t))$ to (30) is used to measure the energy consumption for dropping tasks, which not contains in the formula of total energy consumption (14) ... $\mathbf{1}(\cdot)$ ... is satisfied the following constraint:

$$\alpha^L(t) \cap \alpha^O(t) \cap \alpha^d(t) = \emptyset, t \in \mathcal{T}, \tag{6}$$

$$0 \leq P(t) \leq B(t) < \infty, t \in \mathcal{T}. \tag{15}$$

We obtain the battery energy level evolves as follows:

$$B(t+1) = B(t) - P(t) + E_H(t), t \in \mathcal{T}. \tag{16}$$

Note that the problem of task assignment scheduling is more complicated due to the time-coupled battery constraint. Moreover, different from the existing works, we investigate the problem operate with discrete time slots. We cannot give the expression of actual execution delay of each task as in [16] and [17]. Furthermore, the assumptions of our proposed system model about task decomposition and dropping tasks make our model unsuitable for using Little's law [18].

D. The Execution Delay

The execution delay of tasks experiences consists of two parts, queueing delay at buffer queues and processing delay at processor (local CPU or MEC server). The local processing delay is equal to the local CPU computing time. The MEC server processing delay consists of the offloading transmit time, the MEC server computing time and the downloading transmit time. At the beginning of time slot t, the ith application generates the execution tasks with data size $A_i(t)$ (in bits). The arrived execution tasks are stored as input in task queue $Q_i(t)$ waiting for assignment. The $D_i(t)$ bits of data are allocated as output for processing at the end of each time slot. The output $A_i^D(t)$ bits of return queue are downloaded by mobile device at the end of each time slot. The amount of data $(A_i(t)+Q_i(t)+Q_i^D(t))$ to be processed by the system at t time slot will be executed in $(t, t+T'_{d,i}(t))$ time, and equal to the sum of output data in $(t, t+T'_{d,i}(t))$ time. Thus, we obtain the expression of the actual execution delay as follow:

$$\sum_{\tau_i=t}^{t+T'_{d,i}(t)} \{D_i(\tau_i) + A_i^D(\tau_i)\} = A_i(t) + Q_i(t) + Q_i^D(t). \tag{17}$$

Note that the arrived tasks $A_i(t)$ may be executed at several time slots with different execution ways. The allocated data size $D_i(\tau_i)$ will change with the different time slot. In particular, there is a return of task results $A_i^D(t)$ from return queue $Q_i^D(t)$ to mobile device in MEC server execution. The total execution delay for all execution tasks of applications arrived at time slot t can be expressed as

$$T'_D(t) = \sum_{i=1}^{N} T'_{d,i}(t). \tag{18}$$

E. Problem Formulation

We formulate the problem that minimizes the average weighted sum of energy consumption and execution delay under the condition of battery capacity and queues stability constraints. According to the tasks assignment scheduling, the mobile device makes processing decision at the beginning of each time slot, i.e., local execution, MEC server execution and dropping tasks. The problem is formulated as

$$\mathcal{P}_1 : \min_{\boldsymbol{\alpha}(t), \mathbf{f}(t), \mathbf{P}_{tx}(t), E_H(t)} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{P(t) + \rho \cdot T'_D(t)\}$$

$$s.t.\ (6), (12)-(13), (15), (17)$$

$$P(t) \leq E_{\max}, t \in \mathcal{T} \tag{19}$$

$$0 \leq f_i(t) \leq f_{\max}, i \in \mathcal{N}, t \in \mathcal{T} \tag{20}$$

$$0 \leq p_j(t) \leq p_{tx}^{\max}, j \in \{O, D\}, i \in \mathcal{N}, t \in \mathcal{T} \tag{21}$$

$$\lim_{t\to\infty} \frac{\mathbb{E}[\|\overline{Q}\|]}{t} = 0, i \in \mathcal{N},$$

功率受限

where ρ is the control parameter for implementing the weight on the execution delay in the overall system energy consumption. (21) is the limitation of transmission consumption, where $p_{tx}^{\max}$ is the maximum transmission energy consumption of application i. In each time slot t, the mobile device should dete[r]... ...PU and the ... o deal

$$\overline{Q} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\{Q_i(t) + Q_i^D(t)\}, \tag{5}$$

with the great deal of inconclusive sate information for the harvesting energy, wireless channel, task buffers and results return buffers to make offloading decision. Moreover, $D_i(\tau_i)$ in equation (17) is not precisely tractable, we will use the estimated value of $D_i(\tau_i)$ in our proposed solution. To solve the stochastic optimization problem $\mathcal{P}_1$, we give an improved version of the original problem in the next section.

III. Online Tasks Scheduling Policy

In this section, to find an efficient solution of the primal problem $\mathcal{P}_1$, we present the modified execution delay of our proposed MEC system in the following. We introduce the perturbation parameter θ of battery to simplify the problem. We propose the ODLOO algorithm to obtain the tasks assignment decision and the optimization of the harvested energy. We also analyze the performance of the MEC system.

A. The Modified Execution Delay

When the arrived tasks $A_i(t)$ are executed in several time slots, the actual execution delay $T'_{d,i}(t)$ of $A_i(t)$ may covers some incomplete time slots, making it difficult to find the output data size $D_i(\tau_i)$ of task queue $Q_i(t)$ precisely. We show the relaxed version $T_{d,i}(t)$ of the expression of the execution delay as follow:

$$\begin{aligned} T_{d,i}(t) &= T_{p,i}(t) + T_{q,i}(t) \\ &= \min\left\{\frac{Q_i(t)+A_i(t)}{D_{l,i}(t)+D_{o,i}(t)} + \frac{Q_i^D(t)}{A_i^D(t)}, T_D^{\max}\right\}, t \in \mathcal{T}, \end{aligned} \tag{23}$$

where the $T_{p,i}(t)$ is the processing delay of execution takes $A_i(t)$ in time slot t, $T_{q,i}(t)$ is the queueing delay. In addition, $\min\{\cdot, T_D^{\max}\}$ means that we have a upper bound $T_D^{\max}$ of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

execution delay when the output of queues is small. Note that the tasks $A_i(t)$ can be processed by one of three execution approaches (local execution, MEC execution, or dropping) in the same time slot t. That is, in the denominator $D_{l,i}(t) + D_{o,i}(t)$ of the first term in (23), only one has a value at the same time slot t. In particular, when the tasks are dropped in time slot t, we have $D_{l,i}(t) + D_{o,i}(t) = 0$, marking the first item in above formula to infinity, then we obtain the execution delay for dropping tasks of $T_{d,i}(t) = T_D^{\max}$.

Moreover, the queueing delay for MEC execution consists of two parts, task queueing delay (the first half of the first term of (23)) and return queueing delay (the second term of (23)) respectively. The computing time at MEC server is viewed as a constant, the offloading transmit time is the major delay of MEC server processing delay, which is included in the second half of the first term of the above formula. We assume that the downloading transmit time of data results is approximated as zero. The local execution delay includes queueing delay and processing delay, and the local processing delay is equal to the local CPU computing time. And we obtain the total execution delay of the system of $T_D(t) = \sum_{i=1}^{N} T_{d,i}(t)$.

B. Asymptotic Optimality

We note that the energy limitation of formula (15) makes the system time-dependent, which is a challenge problem for tasks assignment and energy harvesting. We introduce a non-zero lower bound $E_{\min}$ as the minimum discharged energy of battery at each time slot to modify the constraints of problem $\mathcal{P}_1$, and we obtain the modified version as

$$\begin{aligned} \mathcal{P}_2 : & \min_{\boldsymbol{\alpha}(t),\mathbf{f}(t),\mathbf{P}_{tx}(t),E_H(t)} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{P(t) + \rho \cdot T_D(t)\} \\ & \text{s.t. } (6), (12) - (13), (15), (19) - (23) \\ & \quad P(t) \in \{0\} \cup [E_{\min}, E_{\max}], t \in \mathcal{T}, \end{aligned} \tag{24}$$

since the constraint condition of $\mathcal{P}_2$ is more tightened than $\mathcal{P}_1$, then the optimal solution of $\mathcal{P}_2$ must be the greater solution of $\mathcal{P}_1$. By making $E_{\min}$ close to zero, the optimal solution of $\mathcal{P}_2$ will be close to optimal solution of $\mathcal{P}_1$ [17]. Moreover, there are two important parameters that are virtual energy queue and perturbation parameter of battery.

Definition 1: We denote that $\widetilde{B}(t) = B(t) - \theta$, where $\widetilde{B}(t)$ is the virtual energy queue, and it represents the actual energy level of battery that mobile device can be consumed.

Theorem 1: We denote θ is the perturbation parameter of the battery, and the bound constant satisfying

$\theta \geq \widetilde{E} ... V ... E^{-1} ... V ... T^{\max} ... E^{-1}$ (25)

where ... introd... to in... execu... V_t = ... of loc... consu... contai...

which not contains in the formula of total energy consumption (14) ...

$\mathbf{1}(\cdot)$ $\alpha^L(t) \cap \alpha^O(t) \cap \alpha^d(t) = \emptyset, t \in \mathcal{T},$ (6)

is satisfied the following constraint:

$$0 \leq P(t) \leq B(t) < \infty, t \in \mathcal{T}. \tag{15}$$

We obtain the battery energy level evolves as follows:

$$B(t+1) = B(t) - P(t) + E_H(t), t \in \mathcal{T}. \tag{16}$$

Note that ... ore complicate $\sum_{\tau_i = t}^{t+T'_{d,i}(t)} \{D_i(\tau_i) + A_i^D(\tau_i)\} = A_i(t) + Q_i(t) + Q_i^D(t).$ (17) ...re- over, differe... from the existing works, we investigate the problem one... $0 \leq E_H(t) \leq E_H^{\max}, t \in \mathcal{T}.$ (12) cannot give the

C. The ODLOO Algorithm

In this subsection, we introduce the auxiliary optimization function $\mathcal{P}_3$ based on Lyapunov Optimization approach to further optimize the formulated problem. We first define the Lyapunov function as follows:

$$L[\Theta(t)] \triangleq \frac{1}{2}\Big\{\sum_{i=1}^{N}\left[\{Q_i(t)\}^2 + \{Q_i^D(t)\}^2\right] + \widetilde{B}(t)^2\Big\}, \tag{26}$$

where $\widetilde{B}(t) = B(t) - \theta$, and the drift $\Delta(\Theta(t))$ is defined as:

$$\Delta(\Theta(t)) \triangleq \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t) \mid \mathcal{Q}(t), \mathcal{Q}^D(t), \widetilde{B}(t)\}. \tag{27}$$

In order to keep $L[\Theta(t)]$ stable, we should procure the smallest $\Delta(\Theta(t))$. To facilitate the analysis, we obtain the upper bound of $\Delta(\Theta(t))$, which is expressed as

$$\begin{aligned} \Delta(\Theta(t)) \leq\ & \mathbb{E}\Big\{\sum_{i\in\alpha^L(t)}[Q_i(t)(A_i(t) - D_{L,i}(t)) - Q_i^D(t)A_i^D(t)]\Big\} \\ & + \mathbb{E}\Big\{\sum_{i\in\alpha^O(t)} Q_i(t)[A_i(t) - D_{O,i}(t)] + Q_i^D(t)[D_{O,i}(t) \\ & - A_i^D(t)] + \widetilde{B}(t)[E_H(t) - p(t)]\Big\} + \Phi \triangleq \mathbb{E}\{\xi\} + \Phi, \end{aligned} \tag{28}$$

for the sake of writing equation conveniently, we define $\Phi \triangleq \frac{1}{2}\sum_{i=1}^{N}\{[A_i(t) - D_i(t)]^2 + [A_i^C(t) - A_i^D(t)]^2 + R\}$, where $R = \frac{1}{2}[(E_H^{\max})^2 + (\widetilde{E}_{\max})^2]$ is a constant.

Proof: Please refer to Appendix D. ■

Moreover, if the rate of arrived tasks and system processing of each queue can be bounded, the Φ is bounded, then the system can operate stably. The drift-plus-penalty is

$$\begin{aligned} & \Delta(\Theta(t)) + V_p\mathbb{E}\{P(t)\} + V\mathbb{E}\{\rho \cdot T_D(t)\} \\ & = \Delta(\Theta(t)) + V_p\mathbb{E}\{P(t)\} + V_t\mathbb{E}\{T_D(t)\} \\ & \leq \Phi + \mathbb{E}\{\xi + V_pP(t) + V_tT_D(t)\}. \end{aligned} \tag{29}$$

We acquire the minimum value, with $\{\xi + V_pP(t) + V_tT_D(t)\}$ instead of drift-plus-penalty. Therefore, we have the equivalent objective function $\mathcal{P}_3$ as follow:

$$\begin{aligned} \mathcal{P}_3 : & \min_{\boldsymbol{\alpha}(t),\mathbf{f}(t),\mathbf{P}_{tx}(t),E_H(t)} \{\xi + V_pP(t) + V_tT_D(t)\} \\ & = \min\Big\{\sum_{i=1}^{N} Q_i(t)A_i(t) + \widetilde{B}(t)[E_H(t) - P(t)] \\ & - \sum_{i=1}^{N} Q_i^D(t)A_i^D(t) - \sum_{i\in\alpha^L(t)} Q_i(t)D_{L,i}(t) \\ & - \sum_{i\in\alpha^O(t)} D_{O,i}(t)[Q_i(t) - Q_i^D(t)] + V_pP(t) + V_tT_D(t) \end{aligned}$$

each time slot, i.e., local execution, MEC server execution and dropping tasks. The problem is formulated as

$$\begin{aligned} \mathcal{P}_1 : & \min_{\boldsymbol{\alpha}(t),\mathbf{f}(t),\mathbf{P}_{tx}(t),E_H(t)} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{P(t) + \rho \cdot T'_D(t)\} \\ & s.t. (6), (12) - (13), (15), (17) \\ & P(t) \leq E_{\max}, t \in \mathcal{T} \quad (19) \\ & 0 \leq f_i(t) \leq f_{\max}, i \in \mathcal{N}, t \in \mathcal{T} \quad (20) \\ & 0 \leq p_j(t) \leq p_{tx}^{\max}, j \in \{O, D\}, i \in \mathcal{N}, t \in \mathcal{T} \quad (21) \\ & \lim_{t\to\infty} \frac{\mathbb{E}[|\overline{Q}|]}{t} = 0, i \in \mathcal{N}, \end{aligned}$$

功率受限

where ρ is the control parameter for implementing the weight

(30)

...n $\mathcal{P}_3$ consists ...timization of ...variables are ...vo parts (H_1 ...We show the ...ed algorithm

The optimization of EH: We drive the optimal value of the arrived harvested energy by solving the following function:

$$H_1: \min_{0 \leq E_H(t) \leq E_H^{\max}} \widetilde{B}(t) E_H(t), \tag{31}$$

there are two kinds of results about the problem. When $\widetilde{B}(t) \leq 0$, in order to get the smallest H_1, we must obtain the maximum $E_H(t)$, we have the optimal solution as $E_H^*(t) = E_H^{\max} \cdot \mathbf{1}\{\widetilde{B}(t) \leq 0\}$. Further more, when $\widetilde{B}(t) > 0$, similar to the above analysis, we acquire the optimal solution of H_1 by making the value of $E_H(t)$ approaches to zero.

The optimization of tasks assignment: We should solve the equation of $\mathcal{P}_3$ except the term of H_1 to obtain the optimal solution of tasks assignment, we formulate it as follow:

$$\begin{aligned} H_2: & \min_{\boldsymbol{\alpha}(t), \mathbf{f}(t), \mathbf{P}_{tx}(t)} \{\xi + V_p P(t) + V_t T_D(t)\} \\ & \text{s.t. } (6), (19) - (23), (24) \end{aligned} \tag{32}$$

specifically, we denote the takes assignment operation of the system as $\mathbf{f}(t) \triangleq [f_1(t), ..., f_N(t)]$ and $\mathbf{P}_{tx}(t) \triangleq [P_{tx,1}(t), ..., P_{tx,N}(t)]$, which are the set of local frequency and transmission consumption for different application. Further, we obtain the minimum value of H_2 and tasks assignment decision $\boldsymbol{\alpha}(t)$ by choosing the optimal frequency of mobile device and transmission consumption.

Algorithm 1 The Online Dynamic Lyapunov Optimization-based Offloading Algorithm (ODLOO)

??

1: Obtain the current status of $\widetilde{B}(t)$, $\{Q_i(t)\}$ and $\{Q_i^D(t)\}$ at the beginning of the time slot t;
2: **for** $t = 0$ to T **do**
3: Choose $0 \leq f_i(t) \leq f_{\max}$ and $0 \leq p_j(t) \leq p_{tx}^{\max}$ to minimize the values of the target equation of f_l (local execution only), f_o (MEC server computation only), f_d (dropping tasks), respectively;
4: Determine $E_H(t)$ by solving

$$\begin{aligned} \mathcal{P}_3: & \min_{E_H(t)} \{\xi + V_p P(t) + V_t T_D(t)\} \\ & \text{s.t. } (12) - (13) \end{aligned}$$

5: To obtain the tasks assignment decision $\boldsymbol{\alpha}(t)$ ($i \in \alpha^L(t)$, $\alpha^O(t)$ or $\alpha^d(t)$) by execute $\min\{f_l, f_o, f_d\}$;
6: Update the virtual energy queue, task queue and data results queue according to (3), (4) and (16);
7: Set $t = t + 1$.
8: **end for**

D. Performance Analysis

In this subsection, we provide the theoretical proofs of MEC system about the upper bound of battery and the limitation of the energy consumption, as well as the stability of queues.

Theorem 2: According to the optimization, we demonstrate that the upper and lower limits of the battery level $B(t)$, and it confines within $[0, \theta + E_H^{\max}]$, $\forall t \in \mathcal{T}$.

Proof: Depended on the optimization of EH (28), we obtain the upper bound of $B(t)$. When $B(t)$ satisfies with $\theta < B(t) \leq \theta + E_H^{\max}$, notice that the battery has the sufficient power at this time. We suppose $E_H^*(t) = 0$ for obtaining the upper limitation of the battery, and $B(t+1) \leq B(t) \leq \theta + E_H^{\max}$. When $B(t) \leq \theta$, the battery should be charged, we suppose $E_H^*(t) = E_H^{\max}$, obtain $B(t+1) \leq B(t) + E_H^{\max} \leq \theta + E_H^{\max}$. In conclusion, we have the upper bound of battery energy level $B(t) \in [0, \theta + E_H^{\max}]$, $\forall t \in \mathcal{T}$. ■

Theorem 3: For facilitating the analysis, we define the average energy consumption of the system under *Algorithm 1* as p, and obtained the bound $p \leq p^* + \Phi^*/V_p + (V_t/V_P)T_D^*$, where p^* and T_D^* are the optimal solution about energy consumption and execution delay of the problem $\mathcal{P}_1$, respectively. And $\Phi^* = \Phi + \sum_{i=1}^{N} Q_i(t)^{\max} A_i(t)^{\max}$.

Proof: Please refer to Appendix B. ■

Theorem 4: We obtain the upper bound of the queue length of task queue and return queue, respectively. The upper bound of return queue is expressed as $Q_{\max}^D \leq (1/A_{\max}^D)\left\{[\theta + V_p]P_D^{\max} + V_t\left\{T_D^{\min} - T_D^{\max}\right\}\right\} + A_{\max}^D$. The upper bound of task queue is written as $Q_{\max} \leq (1/D_{\max})\{[\theta + V_p]\widetilde{E}_{\max} + V_p \cdot \phi \cdot \mathbf{1}\left(i \in \alpha^d(t)\right) + V_t\{T_D^{\min} - T_D^{\max}\} + Q_{\max}^D D_o^{\max}\} + A_{\max}$.

Proof: Please refer to Appendix C. ■

The proposed ODLOO algorithm is an online algorithm, which only needs the systems current state information as input and does not need the prior systems dynamic information. The character of the algorithm makes it easy to execute for practical MEC system with low complexity. Notice that *Theorem 2* and *Theorem 4* show us the MEC system with a stable battery energy level and queue backlogs. Moreover, we investigate an asymptotic optimization problem.

Theorem 3 exhibits the solution gap between *Algorithm 1* and $\mathcal{P}_1$. The gap decreases with the increasing of control parameter V_p, and increases with control parameter V_t. Thus, we can choose a larger V_p and a smaller V_t to obtain a smaller gap. However, with V_p increasing, more emphasis put on energy saving to solve the formulated $\mathcal{P}_1$. Then, it requires more battery capacity and generates a larger backlog of queues. Thus, we give the appropriate values of V_p and V_t to satisfy the requirements of the task in the practical implementation.

TABLE I: LIST OF SIMULATION DATA PARAMETER

parameter	numerical value	unit
a	0.1	-
b	0.025	-
ϕ	0.1	-
κ	7.8×10^{-21}	-
L_i^{-1}	6400	cycle/bits
$f_{\max}$	16.8×10^9	Hz
$P_{o\max}$	2	mW
ω	1×10^6	Hz
H_i	-50 - -75	dB
N_0	-174	dBm/Hz

IV. Simulation Results

In our simulations, we verify the correctness of the theoretical results in *Section III*, and evaluate the performance of the system by using the Gurobi optimization tool in Matlab. We consider five applications running in the mobile device, each task of applications can be offloaded to the MEC server. In each time slot, the arrived tasks $A_i(t)$ is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

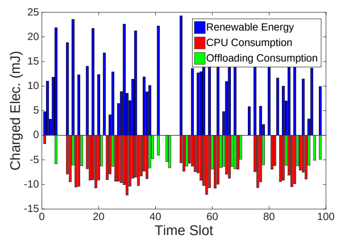


Fig. 2: Elec. charge/discharge ($V_p = 100$).

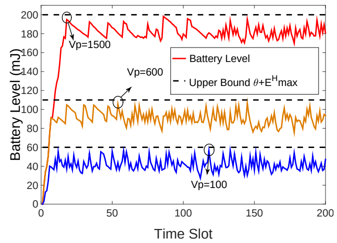


Fig. 3: Battery energy vs. T.

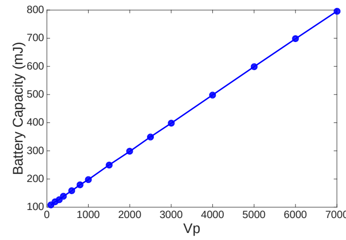


Fig. 4: Battery Capacity vs. V_p.

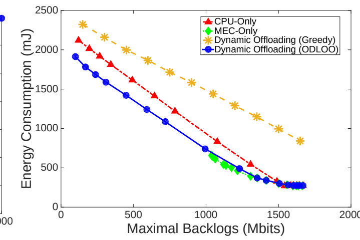


Fig. 5: Energy consumption vs. maximal backlogs.

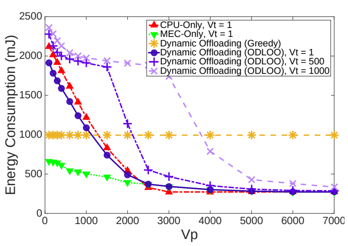


Fig. 6: Energy consumption vs. V_p.

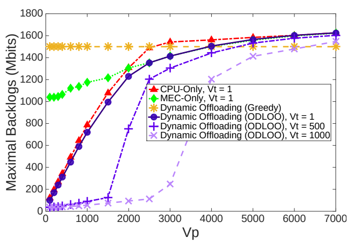


Fig. 7: Maximal backlogs vs. V_p.

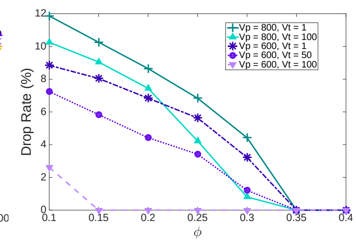


Fig. 8: Drop rate vs. ϕ.

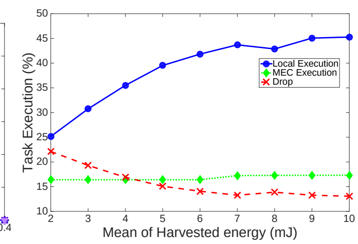


Fig. 9: Task Allocation vs. Mean of Harvested Energy.

uniformly distributed between 0 and 5 Mbits. The harvested energy $E_H(t)$ is uniformly distributed between 1 and 25 mJ. The system has three forms of energy consumption, which are the local energy consumption, the transmission energy consumption for data offloading and downloading. For local execution, $\kappa = 7.8 \times 10^{-21}, \tau = 1s, L_i^{-1} = 6400$ cycle/bits, the maximum value of the frequency is $f_{\max} = 16.8 \times 10^9$ Hz. For MEC server execution, for the uplink, $\omega = 10^6$ Hz, $H_{\max} = -50$ dB, $H_{\min} = -75$ dB, and $H_i(t)$ is uniformly distributed between the two values. The maximum value of uplink transmission energy consumption is $P_{o\max} = 2$ mW. For downlink, the downloaded energy consumption is described as $P_D = a \cdot \omega_D(t) + b$, where $a = 0.1, b = 0.025$. The weight of energy consumption for dropping tasks as $\phi = 0.1$. We set the execution delay for dropping tasks as $T_{\max}^D = 20s$. The experiment simulates for 200 time slots for different V_p. We list the simulation parameter values in Table I.

We first specify the charged renewable energy, execution consumption of local CPU and transmission consumption under $V_p = 100$ in Fig. 2, which matches with the battery level curve in Fig. 3, i.e., the battery level changes along with the amount of charged and discharged electricity. Moreover, the amount of charge and discharge is basically balanced under the long time evolution, which show the stability of the battery, more illustrated the feasibility of the system. In Fig. 3, we show the feasibility of the perturbation parameters θ of battery under $V_p = [100, 600, 1500]$ and verify the upper bound of battery level. We see that the battery level first rises rapidly due to continues charging with renewable resources. Ultimately, the battery level gets stable and fluctuates near the dotted line, which is the upper bound $\theta + E_{\max}^H$. We obtain different θ under the different V_p, and θ increases with V_p. In other words, a larger V_p will lead to a higher battery level. we increase V_p from 1000 to 7000 to investigate the impacts on the battery capacity with different V_p. Fig. 4 shows us the relationship of battery capacity versus V_p. As V_p increases, the battery capacity becomes larger. In addition, the curve depicts the linear relationship between the battery capacity and V_p, which is also mentioned in *Theorem 1* and *Theorem 2*.

Fig. 5 shows energy consumption versus maximal backlogs under the case of Dynamic Offloading (ODLOO), CPU-Only, MEC-Only and Dynamic Offloading (Greedy) policies. We see that our proposed algorithm consumes less energy, while having the same backlogs. Compared with MEC-Only, our ODLOO algorithm has the less backlogs. Fig. 6 and Fig. 7 depict energy consumption versus V_p and maximal backlogs versus V_p under $V_t = 1$ in our proposed algorithm, respectively. Since V_p and V_t affect the tradeoff of system's performance, a small value of V_p results a large amount of energy consumption but a small quantity of backlogs. With V_p increasing, more emphasis put on energy saving. Consequently, the energy consumption curve declines and converges to the optimal value of $\mathcal{P}_1$, while the maximal backlogs curve ascends. Therefore, *Theorem 3* and *Theorem 4* are verified. Otherwise, a small value of V_t results a large amount of backlogs but a small energy consumption. With V_t increasing, more emphasis put on execution delay saving.

Furthermore, we compare the Dynamic Offloading (ODLOO) to CPU-Only, MEC-Only and Dynamic Offloading (Greedy) policies under $V_T = 1$ in Fig. 6 and Fig. 7. When the value of V_p is very small, each algorithm deals with as much arrived tasks as possible. MEC-Only policy has the minimum energy consumption while resulting in a large delay due to its low transmission consumption and large transmission delay. On the contrary, CPU-Only policy has a larger energy consumption and a small delay duo to its high processing speed CPU. Compared to CPU-Only policy, our proposed algorithm consumes less energy, while having the same processing ability due to its dynamic offloading ability. Moreover, since local execution consumes more than data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

transmission, more local execution is cut down to minimize the value of objective function as V_p increases, which makes the curves of CPU-Only policy steepest in Fig. 6 and Fig. 7. The energy consumption curve of the proposed algorithm has a similar dropping tendency as the one of CPU-Only. Moreover, compared with Dynamic Offloading (Greedy) police, our algorithm implements the tradeoff between energy consumption and delay by adjusting the introduced parameter V_p to accommodate different tasks' demand. Note that our proposed algorithm also achieves the low latency and high energy efficiency for MEC system.

We demonstrate the relationship between drop rate and ϕ under $V_p = [600, 800]$ and $V_t = 1$, $V_t = [1, 50, 100]$ and $V_p = 600$ in Fig. 8, respectively. When $V_t = 1$, as V_p increases, i.e., the value of the term $V_p \cdot \phi \cdot \mathbf{1}(\alpha^d(t) = 1)$ in the objective function becomes larger, the drop rate rises sharply. Similarly, when $V_p = 600$, the $V_t T^D_{\max}$ becomes larger with the increase of V_t, the drop rate decreases. Also, we observe that the drop rate decreases with ϕ in the same V_p or V_t, which is consistent with our common sense that ϕ is the weight of dropping tasks.

Furthermore, we simulate for mean values of the harvested energy from 2mJ to 10mJ. As Fig. 9 shows, the rate curves of local execution, MEC execution and dropping indicate the task allocation of the proposed algorithm versus different harvesting conditions. When the mean of E_H is small, the algorithm prefers to either execute tasks locally or drop them for energy savings due to the poor harvesting condition. With the increase of the mean, more harvested energy is available. Therefore, the rate of local execution, which is the most efficient and energy-consuming, grows rapidly, while the drop rate shrinks towards zero. It is crucial for the MEC system to make tasks assignment strategy depends on the mean of harvested energy.

V. Related Works

Communication and computation are widely considered as the higher priorities for future wireless application services, it's crucial to utilize new technologies to increase the effectiveness of communications and achieve low latency , and high energy efficiency in communications [19]. With the ever-increasing intensive computing, computation offloading is considered as an effective way to improve the service performance of mobile devices in terms of both latency and energy consumption [20]. Furthermore, the data transmission occupies the major energy consumption and delay compared with calculation and storage [21]. Moreover, it's also important to achieve high energy efficiency and low latency of data transmission [22]. In [21], a device-to-device (D2D) data transmission method is proposed for solving the limited battery power and storage resources of mobile devices, and it does not need information of task demand, content distribution, and device mobility. Compared with traditional data transmission methods that rely on base stations in a cellular network, D2D achieves high efficiency and low latency data transmission without access point [23], [24]. In addition, different calculation methods also lead to different transmission delays [25].

With the increasing computing energy consumption in the Internet of Things, the high energy efficiency and green computing communication are a common pursuit of academia and industry [26]. In [27], it implemented the green and effective energy management based on the development of energy internet (EI). In addition, EH technology harvests the surrounding renewable resources to provide the major power for MEC systems [28], provides support for the sustainable operation of mobile devices [29]. Some recent studies [30]-[33] focus on minimizing energy consumption or execution delay in MEC system, or to make tradeoff between the two indicators. In [30], the offloading decision and communication resource assignment are optimized to minimize the overall cost of mobile device. In [31], the optimal problem is formulated to save energy consumption of mobile device, which depends on offloading policy (*i.e.*, the arrived tasks are assigned cloud execution or mobile execution). Besides, the Lyapunov optimization-based DREAM algorithm is proposed to minimize CPU and network energy consumption under the execution delay constraints in [32]. The scheduling for offloading and downloading decision is formulated to make the tradeoff between energy consumption and execution delay in MEC system in [33].

Moreover, MEC system also takes EH devices into consideration to improve its service capabilities recently in [17], [34]. A dynamic computation offloading policy is proposed for MEC system with energy harvesting equipment, which moderate the finite battery power as well as limited processing ability of mobile devices in [17]. In [34], according to balancing geographical load, controlling data traffic, and equipping the energy-harvesting devices, the GLOBE algorithm is developed to improve its system performance.

Existing literature focus on optimizing the basic performance parameters of MEC system, i.e., transmit power, bandwidth, computation resource and buffer queues (*i.e.*, offloading queues and downloading queues) [32]-[36]. In [35], the energy-efficient MEC-WPT system is designed by optimizing transmit power, CPU frequencies and tasks offloading. In [36], it optimizes the allocation of bandwidth and obtains the minimum energy consumption with the stability of the task buffers in multi-users MEC system. In [37], it proposes the multi-queue interlacing peak strategy to balance workload by assigning the computing resources. In [38], it focus on allocating computation and communication resource in MEC system. Based on TDMA and OFDMA, the resources of the MEC system are allocated to achieve high energy efficiency and low latency of task offloading [39].

In this paper, we utilize the computation offloading mechanism to mitigate the intensive workload of mobile devices, we make full use of renewable resources that implement the low latency and energy-efficient green communications. Further more, we exploit the tradeoff between energy consumption and execution delay. We can adjust the introducing control parameter to accommodate the requirements of different tasks, which improves system service capabilities and user's experience greatly, especially for delay-sensitive tasks. Compared to the existing studies, it is complicated to implement the tasks assignment (load balancing and downloading scheduling) in the MEC system with EH devices and data buffers.

VI. Conclusions

In this paper, we investigated the energy-delay tradeoff for dynamic offloading in mobile-edge computing (MEC) system with energy harvesting (EH) devices. With the buffer queue and battery energy level stable, we proposed a problem of minimizing energy consumption. We proposed an online dynamic Lyapunov optimization-based offloading (ODLOO) algorithm to make decisions for tasks assignment. Performance analysis shown the tradeoff between energy and delay, as well as the asymptotic optimality of proposed algorithm. The simulation results verified the correctness of theories, and demonstrated that our proposed algorithm implemented high energy efficient and low latency compared to CPU-Only policy.

Acknowledgment

This work is supported by the NSF of China under Nos. 61772130, 61473189, 61301118; the International S&T Cooperation Program of Shanghai Science and Technology Commission under No. 15220710600; and the Natural Science Foundation of Shanghai (No. 17ZR1445200).

Appendix

A. Proof of Theorem 1

According to the simplification of objective function (30), we obtain the target equation of local execution, the target equation of MEC server execution and the target equation of dropping tasks, respectively.

Local Execution: Due to the tasks only be executed by mobile device, we obtain $\alpha^d(t)=0$, $D_{O,i}(t)=0$, $Q_i^D(t)=0$ and $P_1(t)\triangleq P(t)=P_L(t)$, then we have the target equation of local execution as follow:

$$\begin{aligned} F_L &\triangleq \min_{\mathbf{f}(t),E_H(t)}\{\xi+V_pP(t)+V_tT_D(t)\} \\ &=\min\{\sum_{i=1}^{N}Q_i(t)[A_i(t)-D_{L,i}(t)]+\widetilde{B}(t)E_H(t) \\ &\quad+[V_p-\widetilde{B}(t)]P_L(t)+V_tT_D(t)\}. \end{aligned} \tag{33}$$

MEC server Execution: When the arrived tasks are offloaded to the MEC server only, we have $\alpha^d(t)=0$, $D_{L,i}(t)=0$, and $P_2(t)\triangleq P(t)=P_O(t)+P_D(t)$, thus, we obtain the following target equation of offloading to cloud server:

$$\begin{aligned} F_O &\triangleq \min_{\mathbf{P}_{tx}(t),E_H(t)}\{\xi+V_pP(t)+V_tT_D(t)\} \\ &=\min\{\sum_{i=1}^{N}Q_i(t)[A_i(t)-D_{O,i}(t)]+\widetilde{B}(t)E_H(t) \\ &\quad+\sum_{i=1}^{N}Q_i^D(t)[D_{O,i}(t)-A_i^D(t)]+V_tT_D(t) \\ &\quad+[V_p-\widetilde{B}(t)][P_O(t)+P_D(t)]\}. \end{aligned} \tag{34}$$

Dropping tasks: According to the assumption, when the tasks are dropped, the entire system will stop working. And we assume that the size of dropping takes is the same as the arrived tasks data. It is worth noting that ϕ is the weight of dropping tasks energy consumption and will not affect the battery power. Then we have $i\in\alpha^d(t)$, and the target equation of dropping tasks is described as

$$\begin{aligned} F_d &\triangleq \min_{E_H(t)}\{\xi+V_pP(t)+V_tT_D(t)\} \\ &=\min\left\{\widetilde{B}(t)E_H(t)+V_p\cdot\phi\cdot\mathbf{1}(i\in\alpha^d(t))+V_tT_D^{\max}\right\} \end{aligned} \tag{35}$$

There are three calculation methods in the system for mobile device to choose at the beginning of each time slot t. According to the ODLOO algorithm, the mobile device selects the optimal method of minimum energy consumption to determine the operation decision (i.e., the arrived tasks are executed locally, or offloaded to MEC server, or dropped.). We assume that the current energy of battery is just only sufficient to supply the current system operation, the mobile device will make the decision to drop the tasks to save energy in the next moment. There are two cases of energy consumption comparison, the energy consumption for local execution and dropping tasks, or the energy consumption for MEC server consumption and dropping tasks. According to *Definition 1*, $\widetilde{B}(t)=B(t)-\theta$, which will be brought into the formula in the following proof.

(1) Local execution and dropping : In order to satisfy the condition that the energy of battery only satisfies the current system's energy consumption, $B(t)=P_1(t)$, where $P_1(t)$ is the value of current energy consumption locally. If the tasks will be dropped at the next moment, the target function value of the local calculation should be greater than the value of the target function at the time of dropping tasks, that is $F_L>F_d$. We have the following formula after simplification:

$$\begin{aligned} &\sum_{i=1}^{N}Q_i(t)[A_i(t)-D_{L,i}(t)]+\widetilde{B}(t)E_H(t)+[V_p-\widetilde{B}(t)]P_L(t) \\ &+V_tT_D(t)>\widetilde{B}(t)E_H(t)+V_p\cdot\phi+V_tT_D^{\max}, \end{aligned} \tag{36}$$

we plug $\widetilde{B}(t)=B(t)-\theta$ and $B(t)=P_1(t)$ in the above formula, we derive the equation of θ as follow:

$$\begin{aligned} \theta &> (\frac{\phi}{P_1(t)}-1)V_p+\left(\frac{T_D^{\max}}{P_1(t)}\right)V_t-\frac{V_tT_D(t)}{P_1(t)} \\ &\quad-\frac{\sum_{i=1}^{N}Q_i(t)[A_i(t)-D_{L,i}(t)]}{P_1(t)}+P_1(t) \\ &> P_1(t)+(\frac{\phi}{P_1(t)}-1)V_p+\left(\frac{T_D^{\max}}{P_1(t)}\right)V_t \\ &> P_1^{\max}+(\frac{\phi}{P_1^{\min}})V_p+(\frac{T_D^{\max}}{P_1^{\min}})V_t. \end{aligned} \tag{37}$$

For finding the maximum of the right in the above formula, we leave out items that are greater than zero. Note that the third term is greater than zero, and the forth terms is greater than zero (because $A_i(t)$ is always greater than $D_{L,i}(t)$). we quit the two terms. For taking the maximum value of $P_1(t)$, we have $\theta>k(f_{\max})^2+(\phi/P_1^{\min})V_p$, where $P_1^{\max}=k(f_{\max})^2$ is the maximum value of energy consumed locally, and $P_1^{\min}$ is the minimum energy consumption.

(2) MEC server execution and dropping : Similar to *Local execution and dropping*, at this point, $B(t)=P_2(t)$, where $P_2(t)$ is the current energy consumption of the system. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

order to satisfy the condition of dropping tasks in the next moment, $F_O > F_d$ must be guaranteed. Then, we obtain

$$\begin{aligned}&\sum_{i=1}^{N} Q_i(t)[A_i(t) - D_{O,i}(t)] + [V_p - \widetilde{B}(t)][P_O(t) + P_D(t)]\\&+ \widetilde{B}(t)E_H(t) + \sum_{i=1}^{N} Q_i^D(t)[D_{O,i}(t) - A_i^D(t)] + V_t T_D(t)\\&> \widetilde{B}(t)E_H(t) + V_p \cdot \phi + V_t T_D^{\max},\end{aligned} \tag{38}$$

we plug $\widetilde{B}(t) = B(t) - \theta$ and $B(t) = P_2(t)$ in the above formula (38), we also derive the equation of θ as follow:

$$\begin{aligned}\theta &> (\frac{\phi}{P_2(t)} - 1)V_P + \left(\frac{T_D^{\max}}{P_2(t)}V_t\right) - \frac{V_t T_D(t)}{P_2(t)} + P_2(t)\\&- \frac{\sum_{i=1}^{N} Q_i(t)[A_i(t) - D_{L,i}(t)]}{P_2(t)}\\&- \frac{\sum_{i=1}^{N} Q_i^D(t)[D_{O,i}(t) - A_i^D(t)]}{P_2(t)}\\&> P_2(t) + (\frac{\phi}{P_2(t)} - 1)V_p + \frac{T_D^{\max}}{P_2(t)}V_t\\&> P_2^{\max} + (\frac{\phi}{P_2^{\min}})V_p + \left(\frac{T_D^{\max}}{P_2^{\min}}\right)V_t,\end{aligned} \tag{39}$$

similarly, the fourth and fifth items in (39) are always greater than zero, we obtain the maximum of the right in (39) by removing the two terms. Where $P_2^{\max} = 2P_{tx}^{\max}$ is the maximum energy consumption of offloading the tasks, $P_2^{\min}$ is the minimum power consumption. Based on the above two cases, we have $\theta > \widetilde{E}_{\max} + V_p \cdot \phi \cdot E_{\min}^{-1} + V_t \cdot T_D^{\max} \cdot E_{\min}^{-1}$.

B. Proof of Theorem 3

For the dynamic time-dependent expression of $B(t)$ in formula (16), we expect it and sum it over the period $[0, T-1]$ on both sides of the formula, and for all applications we have

$$\mathbb{E}\{B(T)\} - \mathbb{E}\{B(0)\} = \sum_{t=0}^{T-1}[\mathbb{E}\{E_H(t)\} - \mathbb{E}\{P(t)\}]. \tag{40}$$

Note that the battery is bounded as $E_{\min} \le B(t) \le E_{\max}$, and we obtain the relationship of the battery between charge and discharge under the long time evolution by dividing both sides by T and letting T approach to infinity, then we obtain

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{E_H(t)\} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{P(t)\}, \forall i, \tag{41}$$

it means that the input and output values of the battery are equal under the long time evolution, we obtain the following function after the relaxation of the constraints of $\mathcal{P}_1$,

$$\begin{aligned}\widehat{\mathcal{P}} : \min \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{P(t) + \rho T_D(t)\}, \forall i,\\ \text{s.t. } (6), (12) - (13), (19) - (23), (24).\end{aligned} \tag{42}$$

Since the constraints of $\mathcal{P}_1$ are stricter than the problem $\widehat{\mathcal{P}}$, the optimal solution of $\mathcal{P}_1$ is also applicable to $\widehat{\mathcal{P}}$. We define that the solution of $\widehat{\mathcal{P}}$ as $\widehat{p}$, and obtain $\widehat{p} \le p^*$.

We obtain the minimum optimal solution by exploiting the format of drift-plus-penalty (29) to replace the original problem $\mathcal{P}_1$. Substitute (30) in (29), the R.H.S of (30) becomes

$$\begin{aligned}&\Phi + \sum_{i=1}^{N}\mathbb{E}\{Q_i(t)\widehat{A}_i(t)\} + \mathbb{E}\{\widetilde{B}(t)[\widehat{E}_H(t) - \widehat{P}(t)]\}\\&- \sum_{i=1}^{N}\mathbb{E}\{Q_i^D(t)\widehat{A}_i^D(t)\} - \sum_{i\in\alpha^L(t)}\mathbb{E}\{Q_i(t)\widehat{D}_{L,i}(t)\}\\&- \sum_{i\in\alpha^O(t)}\mathbb{E}\{\widehat{D}_{O,i}(t)[Q_i(t) - Q_i^D(t)]\}\\&+ V_p\mathbb{E}\{\widehat{P}(t)\} + V_t\mathbb{E}\{\widehat{T}_D(t)\},\end{aligned} \tag{43}$$

notice that $\mathbb{E}\{\widehat{E}_H(t) - \widehat{P}(t)\} = 0$. In order to acquire the maximum value of the right part in above formula (43), the three subtraction terms in the second equation are greater than zero, which are subtracted, thus the R.H.S of (29) becomes

$$\Phi + \sum_{i=1}^{N} Q_i^{\max}(t)A_i^{\max}(t) + V_p p^* + V_t T_D^*, \tag{44}$$

where $Q_i^{\max}(t)$ and $A_i^{\max}(t)$ are the maximum data size of the tasks queue and the arrived tasks at time slot t, respectively. Add the above formula from 0 to $T-1$, and let the item contains drift to the left, we have

$$\begin{aligned}&\sum_{t=0}^{T-1}\{V_p\mathbb{E}\{P(t)\} + V_t\mathbb{E}\{T_D(t)\}\} \le \Phi T + T\sum_{i=1}^{N} Q_i^{\max}(t)A_i^{\max}(t)\\&+ TV_p p^* + TV_t T_D^* - [\mathbb{E}\{L(T)\} - \mathbb{E}\{L(0)\}]\\&\le \Phi^* T + TV_p p^* + TV_t T_D^* + \mathbb{E}\{L(0)\},\end{aligned} \tag{45}$$

where $\Phi^* = \Phi + \sum_{i=1}^{N} Q_i^{\max}(t)A_i^{\max}(t)$, the two sides of the above formula are divided by $V_p \cdot T$, and let T tends to infinity, and we obtain $\lim_{T\to\infty}(1/T)\mathbb{E}\{L(0)\} = 0$, therefore, the average energy consumption of the system is written as $\lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}\mathbb{E}\{P(t)\} \le p^* + (\Phi^*/V_p) + (V_t/V_P)T_D^*$.

C. Proof of Theorem 4

According to the objective function $\mathcal{P}_3$, we obtain

$$\begin{aligned}&\xi + V_p\mathbb{E}\{P(t)\} + V_t\mathbb{E}\{T_D(t)\}\\&= \sum_{i=1}^{N}\mathbb{E}\left\{Q_i(t)[A_i(t) - D_{L,i}(t) - D_{O,i}(t)]\right\}\\&+ \sum_{i=1}^{N}\mathbb{E}\left\{Q_i^D(t)[A_i^C(t) - A_i^D(t)]\right\} + \mathbb{E}\{\widetilde{B}(t)[E_H(t) - P(t)]\}\\&+ V_p\mathbb{E}\{P(t)\} + V_t\mathbb{E}\{T_D(t)\} + V_p \cdot \phi \cdot \mathbf{1}(i \in \alpha^d(t)),\end{aligned} \tag{46}$$

in order to find the upper bound of the queue length of task queue and return queue, let the above formula $\xi + V_p\mathbb{E}\{P(t)\} + V_t\mathbb{E}\{T_D(t)\} \ge \sum_{i=1}^{N}[Q_i(t)A_i(t)] + \widetilde{B}(t)E_H(t) + V_t T_D^{\max}$, where the $\sum_{i=1}^{N}[Q_i(t)A_i(t)] + \widetilde{B}(t)E_H(t) + V_t T_D^{\max}$ represents the objective function with no execution. We first obtain the upper bound of the return queue by

$$\begin{aligned}&- \sum_{i=1}^{N}[Q_i^D(t)A_{\max}^D] + \widetilde{B}(t)[-P_D^{\max}] + V_p P_D^{\max} + V_t T_D^{\min}\\&\le V_t T_D^{\max},\end{aligned} \tag{47}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

where $A^D_{\max}$ is the maximum of the downloadable data results, $P^{\max}_D$ is the maximum of the energy consumption of downloading data results, $T^{\min}_D$ is the minimum of the execution delay. As for the following formula (48), suppose the item contains queue length on the one side, set others items on the other side, both sides are divided by $A^D_{\max}$, thus we obtain

$$\sum_{i=1}^{N} Q_i^D(t) \geq \frac{1}{A^D_{\max}}\Big\{\widetilde{B}(t)[-P^{\max}_D] + V_p P^{\max}_D + V_t\left\{T^{\min}_D - T^{\max}_D\right\}\Big\} \geq \frac{1}{A^D_{\max}}\Big\{[\theta + V_p]P^{\max}_D + V_t\left\{T^{\min}_D - T^{\max}_D\right\}\Big\}. \quad (48)$$

Due to the maximum return queue backlog at the moment, more tasks will be executed in the next moment, we have $Q_i^D(t+1) < Q_i^D(t) \leq (1/A^D_{\max})\Big\{[\theta + V_p]P^{\max}_D + V_t\left\{T^{\min}_D - T^{\max}_D\right\}\Big\} + A^D_{\max}$, we obtain the upper bound $Q^D_{\max}$ of return queue as follow:

$$\lim_{T\to\infty} \sup \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\left\{Q_i^D(t)\right\} \leq \frac{1}{A^D_{\max}}\Big\{[\theta + V_p]P^{\max}_D + V_t\left\{T^{\min}_D - T^{\max}_D\right\}\Big\} + A^D_{\max}. \quad (49)$$

Similarly, we obtain the upper bound of the task queue by

$$-\sum_{i=1}^{N}[Q_i(t)D_{\max}] + \sum_{i=1}^{N}[Q_i^D(t)D_{o,i}(t)] + \widetilde{B}(t)[-\widetilde{E}_{\max}] + V_p\widetilde{E}_{\max} + V_p\cdot\phi\cdot\mathbf{1}\left(i\in\alpha^d(t)\right) + V_t T^{\min}_D \leq V_t T^{\max}_D, \quad (50)$$

where $D_{\max} = \max\{D^{\max}_o, D^{\max}_l, D^{\max}_d\}$, and the $D^{\max}_o, D^{\max}_l$, and $D^{\max}_d$ are the maximum of outputs of task queue of the three calculation methods, respectively. We obtain the upper bound $Q_{\max}$ of task queue as follow:

$$\lim_{T\to\infty} \sup \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\left\{Q_i(t)\right\} \leq \frac{1}{D_{\max}}\Big\{[\theta + V_p]\widetilde{E}_{\max} + V_p\cdot\phi\cdot\mathbf{1}\left(i\in\alpha^d(t)\right) + V_t\left\{T^{\min}_D - T^{\max}_D\right\} + Q^D_{\max}D^{\max}_o\Big\} + A_{\max}, \quad (51)$$

where $A_{\max}$ is the maximum of the arrived tasks.

D. Proof for the upper bound function $\Delta(\Theta(t))$

According to (16), (3) and (4), we obtain the following formulas:

- for the battery queue:

$$\{B(t+1)\}^2 \leq \{B(t) - P(t) + E_H(t)\}^2, \quad (52)$$

- for the task queue:

$$\{Q_i(t+1)\}^2 \leq \{Q_i(t) + A_i(t) - D_{\sum,i}(t)\}^2, \quad (53)$$

- for the data results return queue:
  when $i\in\alpha^L(t)$, we have

$$\{Q_i^D(t+1)\}^2 \leq \{Q_i^D(t) - A_i^D(t)\}^2, \quad (54)$$

  and $i\in\alpha^O(t)$, we obtain

$$\{Q_i^D(t+1)\}^2 \leq \{Q_i^D(t) + A_i^C(t) - A_i^D(t)\}^2. \quad (55)$$

Substituting the above four formulas into (27), then we obtain the upper bound function of drift $\Delta(\Theta(t))$ in (28).

According to the expression of Φ, we realize that R is produced in $[\widetilde{B}(t+1)]^2 - [\widetilde{B}(t)]^2$ after simple derivation. Subtract θ on both sides of (16), then on both sides of the square we can have

$$\{\widetilde{B}(t+1)\}^2 \leq \{\widetilde{B}(t) - P(t) + E_H(t)\}^2, \quad (56)$$

and we also obtain

$$\begin{aligned}
&\{\widetilde{B}(t+1)\}^2 - \{\widetilde{B}(t)\}^2 \\
&\leq \{\widetilde{B}(t) - P(t) + E_H(t)\}^2 - \{\widetilde{B}(t)\}^2 \\
&= 2\widetilde{B}(t)[E_H(t) - P(t)] + [P^2(t) + E_H^2(t) - 2P(t)E_H(t)] \\
&\leq 2\widetilde{B}(t)[E_H(t) - P(t)] + \{\widetilde{E}^2_{\max} + [E^{\max}_H]^2\}, \quad (57)
\end{aligned}$$

where $R = \frac{(E^{\max}_H)^2 + (\widetilde{E}_{\max})^2}{2}$. Then, we conclude the proof.

References

[1] A. A. Simiscuka, M. Bezbradica, and G. M. Muntean, "Performance analysis of the quality of service-aware networking scheme for smart internet of things gateways," in *IWCMC*, June 2017, pp. 1370–1374.

[2] K. Wang, J. Mi, C. Xu, L. Shu, and D. J. Deng, "Real-time big data analytics for multimedia transmission and storage," in *IEEE International Conference on Communications in China*, July 2016, pp. 1–6.

[3] K. Kumar, J. Liu, Y. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[4] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Comm.*, vol. 20, no. 3, pp. 46–53, 2013.

[5] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. M. Leung, "Emc: Emotion-aware mobile cloud computing in 5g," *IEEE Network*, vol. 29, no. 2, pp. 32–38, March 2015.

[6] N. K. Shah, "Big data and cloud computing: Pitfalls and advantages in data management," in *International Conference on Computing for Sustainable Global Development*, March 2015, pp. 643–648.

[7] A. Gajbhiye and K. M. P. Shrivastva, "Cloud computing: Need, enabling technology, architecture, advantages and challenges," in *Confluence The Next Generation Information Technology Summit*, Sept 2014, pp. 1–7.

[8] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Rajan, "Optimal decision making for big data processing at edge-cloud environment: An sdn perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, Feb. 2018.

[9] Y. Liu and M. J. Lee, "An effective dynamic programming offloading algorithm in mobile cloud computing system," in *IEEE Wireless Communications and Networking Conference*, April 2014, pp. 1868–1873.

[10] M. Mukherjee, L. Shu, D. Wang, K. Li, and Y. Chen, "A fog computing-based framework to reduce traffic overhead in large-scale industrial applications," in *IEEE INFOCOM WKSHPS*, May 2017, pp. 1008–1009.

[11] H. Jabbar, Y. S. Song, and T. T. Jeong, "Rf energy harvesting system and circuits for charging of mobile devices," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 247–253, February 2010.

[12] S. L. et al, "World-wide electricity consumption of communication networks," in *Optical Express*, vol. 20, no. 26, 2012, pp. 513–524.

[13] F. Shahzad and T. R. Sheltami, "An efficient mac scheme in wireless sensor network with energy harvesting (ehwsn) for cloud based applications," in *IEEE LCN Workshops*, Oct 2015, pp. 783–788.

[14] Y. Mao, Y. Luo, J. Zhang, and K. B. Letaief, "Energy harvesting small cell networks: feasibility, deployment, and operation," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 94–101, June 2015.

[15] L. Huang, "The value-of-information in matching with queues," *IEEE/ACM Trans. on Networking*, vol. 25, no. 1, pp. 29–42, Feb 2017.

[16] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, July 2016, pp. 1451–1455.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2843365, IEEE Transactions on Industrial Informatics

[17] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Selected Areas in Comm.*, vol. 34, no. 12, pp. 3590–3605, 2016.

[18] S. M. Ross, "Introduction to probability models," *Academic Press*, 2014.

[19] B. Liang, "Mobile edge computing," *Key Technologies for 5G Wireless Systems, V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, Eds., Cambridge University Press*, vol. 16, no. 3, pp. 1397–1411, March 2017.

[20] J. L. D. Neto, S. y. Yu, D. F. Macedo, J. M. S. Nogueira, R. Langar, and S. Secci, "Uloof: a user level online offloading framework for mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.

[21] Z. Liang, D. Wu, J. Chen, and Z. Dong, "Greening the smart cities: Energy-efficient massive content delivery via d2d communications," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[22] L. Zhou, D. Wu, J. Chen, and Z. Dong, "When computation hugs intelligence: Content-aware data processing for industrial iot," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[23] D. Wu, L. Zhou, and Y. Cai, "Social-aware rate based content sharing mode selection for d2d content sharing scenarios," *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2571–2582, Nov 2017.

[24] H. Trinh, D. Chemodanov, S. Yao, Q. Lei, B. Zhang, F. Gao, P. Calyam, and K. Palaniappan, "Energy-aware mobile edge computing for low-latency visual data processing," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud*, Aug 2017, pp. 128–133.

[25] L. Zhou, D. Wu, Z. Dong, and X. Li, "When collaboration hugs intelligence: Content delivery over ultra-dense networks," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 91–95, DECEMBER 2017.

[26] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial internet of things architecture: An energy-efficient perspective," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48–54, December 2016.

[27] M. Gao, K. Wang, and L. He, "Probabilistic model checking and scheduling implementation of energy router system in energy internet for green cities," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2018.

[28] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 443–461, Third 2011.

[29] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, March 2015.

[30] M. H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," in *IEEE International Conference on Communications*, May 2016, pp. 1–6.

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

[32] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, Dec 2015.

[33] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306–2316, 2015.

[34] J. Xu, H. Wu, L. Chen, C. Shen, and W. Wen, "Online geographical load balancing for energy-harvesting mobile edge computing," in *Available: https://arxiv.org/abs/1704.00107*, Apr 2017.

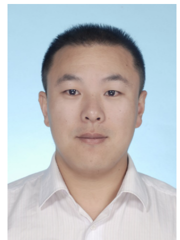
[35] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," in *IEEE International Conference on Communications*, 2017, pp. 1–6.

[36] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *IEEE GLOBECOM*, Dec 2016, pp. 1–6.

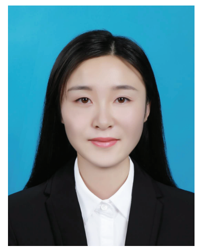
[37] L. Zuo, S. Dong, L. Shu, C. Zhu, and G. Han, "A multiqueue interlacing peak scheduling method based on tasks' classification in cloud computing," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–13, 2016.

[38] M. H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *IEEE INFOCOM*, 2017, pp. 1–9.

[39] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, March 2017.



Guanglin Zhang received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University in 2012. From 2013 to 2014, he was a Postdoctoral Research Associate with the Institute of Network Coding, Chinese University of Hong Kong. He is currently a professor and Department Chair with the Department of Communication Engineering, Donghua University. His research interests include mobile edge computing, content centric networks, vehicular networks, and smart micro-grid. He serves as a TPC Member for IEEE Globecom 2016-2017, IEEE ICC 2014, 2015, 2017, IEEE VTC2017-Fall, and IEEE/CIC ICCC 2014. He serves as the Local Arrangement Chair of ACM TURC 2017, and Vice TPC Co-Chair of ACM TURC 2018. He serves as Associate Editors of China Communications, Journal of Communications and Information Networks, and IEEE Access.



Wenqian Zhang was born in 1994. She received the B.S. degree in Electronic Information Science and Technology from Binzhou University in 2016. She is currently a PhD student with Department of Communication Engineering, College of Information Science and Technology, Donghua University. Her research interests include the energy-delay tradeoff in mobile edge computing, and the tasks offloading decision optimization.

Yu Cao received the B.S. degree in Communication engineering from Donghua University in 2016. He is currently a Master student with Department of Communication Engineering, College of Information Science and Technology, Donghua University. His research interests include green communication, smart grid and mobile edge computing, etc.

Demin Li received the Ph.D. in electronic and computer engineering from Nanjing University of Science and Technology, China, in 1998. He is currently a professor of the Department of Communication Engineering, College of Information Science and Technology, Donghua University, Shanghai, China. From March 2002 to June 2005, he was a research scientist of Nanjing University, China. His recent research interests include telecommunication system engineering, wireless mobile networking, mobile decision theory and mobile decision support systems. He published more than 60 research papers in journals and conferences.

Lin Wang received the Ph.D. degree in control theory from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2009, and then joined the Department of Automation, Shanghai Jiao Tong University (SJTU), China. She was a visiting scholar at the Division of Optimization and Systems Theory, Royal Institute of Technology (KTH), from Apr. 2010 to Sep. 2010 and Sep. 2011 to Feb. 2012. Currently, she is an Associate Professor of SJTU. Her research interests include multi-agent systems, adaptive complex networks, and coordination of multiple manipulators.