

# Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things

Ying Chen, Ning Zhang, *Member, IEEE*, Yongchao Zhang, Xin Chen, Wen Wu, *Student Member, IEEE* and Xuemin (Sherman) Shen, *Fellow, IEEE*

**Abstract**—With proliferation of computation-intensive Internet of Things (IoT) applications, the limited capacity of end devices can deteriorate service performance. To address this issue, computation tasks can be offloaded to the Mobile Edge Computing (MEC) for processing. However, it consumes considerable energy to transmit and process these tasks. In this paper, we study the energy efficient task offloading in MEC. Specifically, we formulate it as a stochastic optimization problem, with the objective of minimizing the energy consumption of task offloading while guaranteeing the average queue length. Solving this offloading optimization problem faces many technical challenges due to the uncertainty and dynamics of wireless channel state and task arrival process, and the large scale of solution space. To tackle these challenges, we apply stochastic optimization techniques to transform the original stochastic problem into a deterministic optimization problem, and propose an energy efficient dynamic offloading algorithm called EEDOA. EEDOA can be implemented in an online way to make the task offloading decisions with polynomial time complexity. Theoretical analysis is given to demonstrate that EEDOA can approximate the minimal transmission energy consumption while still bounding the queue length. Experiments results are presented which shows the EEDOA's effectiveness.

**Index Terms**—Internet of Things; Mobile Edge Computing; Energy Efficient Offloading; Dynamic Offloading



## 1 INTRODUCTION

With the prosperous development of Internet of Things (IoT), the number of computation-intensive applications running on the IoT devices becomes ever-increasing [1], [2]. These computation-intensive applications typically require powerful computing ability and incurs high energy consumption to process them locally on the devices, whereas the computing and battery capacities of the IoT devices are limited. To tackle the challenge, the computation tasks can be offloaded from IoT devices to the cloud with powerful computing capacity for processing. As a promising technology, mobile edge computing (MEC) is introduced to provide computing service at the network edge. Unlike the conventional cloud computing which is remote from the IoT devices, MEC can be deployed at the radio access point such as a base station. MEC can help reduce the traffic of core network and the service latency [3]. The IoT devices can get better computing service and prolong the battery life by offloading the computation tasks [4]. Therefore, the task offloading in MEC for IoT has attracted significant attention from both industry and academia [5], [6].

Computation offloading from the IoT devices to MEC incurs high energy consumption which accounts for a significant portion of the device's total energy consumption [7], [8]. In IoTs, the energy consumption for transmission of each device is greatly affected by the wireless channel state. When the channel condition is better, the transmission rate would be higher, and it needs less time to transmit data, thus reducing the transmission energy consumption. On the contrary, when the channel condition is worse, more energy would be consumed to transmit the same amount of data. Thus, the IoT devices' transmission energy consumption could be reduced by postponing task offloading until the wireless channel becomes better. However, the IoT devices' queue lengths would become large and even unstable [9]. Therefore, it is critical to design an effective task offloading strategy which could optimize the energy efficiency while providing performance guarantees for the IoT devices.

It faces several challenges to design such an effective offloading strategy. Firstly, the quality of the wireless channel is dynamic and changeable over time. It can be affected by not only the locations of IoT devices, but also other factors like network congestion, fading, etc [10]. Moreover, the time-varying channel state is hard to be predicted exactly in practice. Besides, the task arrival process of each IoT device is also hard to obtain. Therefore, it is of great challenge to design an offloading strategy which could adapt to the dynamics of channel condition and task arrival. Secondly, as the IoT devices' number rises rapidly, the scale of the energy efficient dynamic offloading

- Ying Chen, Yongchao Zhang and Xin Chen are with the Computer School, Beijing Information Science and Technology University (BIS-TU), Beijing 100101, China. (E-mail: chenying@bistu.edu.cn, zhangyongchao@mail.bistu.edu.cn, chenxin@bistu.edu.cn)
- Ning Zhang is with Texas A&M University-Corpus Christi, USA. (E-mail: ning.zhang@tamucc.edu)
- Wen Wu and Xuemin (Sherman) Shen are with University of Waterloo, Canada. (E-mail: w77wu@uwaterloo.ca, sshen@uwaterloo.ca)

problem would be huge [11]. Thus, it is vital to design a low-complexity and efficient offloading algorithm to address this issue.

This article focuses on the energy efficient dynamic offloading problem in MEC for IoT. A stochastic optimization problem is formulated aiming at minimizing the average transmission energy consumption and guaranteeing the devices' performance. By taking advantage of stochastic optimization techniques, we design an energy efficient dynamic offloading algorithm (EEDOA) to solve this problem. EEDOA dynamically makes offloading decisions without any statistics of channel condition or task arrival process, and it balances the energy efficiency and queue length by setting a parameter  $V$ . Mathematical analysis is given which demonstrates that the time complexity of EEDOA is polynomial. In addition, EEDOA can approximate the optimal transmission energy consumption while still guaranteeing the upper bound of queue length. Experiment results show the EEDOA's effectiveness.

For the rest of this article, we present related work in Sec. 2. System model and energy efficient offloading optimization problem are presented in Sec. 3. In Sec. 4, EEDOA is proposed to solve this optimization problem effectively. Sec. 5 gives the performance analysis for EEDOA. In Sec. 6, experiments are conducted to evaluate EEDOA. This article is concluded in Sec. 7.

## 2 RELATED WORK

There had been some works about the computation offloading in MEC. Munoz et al. [11] focused on a femto-cloud system and presented an optimal computation and radio resources allocation strategy to optimize the energy consumption and latency concurrently. Wang et al. [12] focused on the energy saving problem in cloud radio access network, and formulated a non-convex energy minimization optimization problem. Then, an iterative algorithm was designed to solve this problem. Chen et al. [13] proposed a distributed algorithm by taking advantage of game theory for the computation offloading problem.

Yu et al. [14] studied the computing and radio resources allocation strategy for the MEC system, where multiple devices accessed the MEC by Orthogonal Frequency-Division Multiplexing Access. They proposed an allocation algorithm which could save energy and offload more computing tasks at the same time. Lyu et al. [15] focused on the task offloading in proximate cloud, and formulated a framework to optimize the system utility. The NP-hard of the proposed utility optimization problem was proven, and a heuristic algorithm was given. You et al. [16] considered the case that the wireless energy transfer powered the mobile cloud computing, and formulated an energy efficient framework to maximize the computing probability with the constraints of energy and

delay. Most of these researches were based on the prediction or assumption of task arrival or channel state. However, the offloading traffic of IoT devices and the wireless channels' quality are highly dynamic and hard to predict exactly.

To deal with this challenge, some stochastic optimization techniques have been applied in recent works. Mao et al. [17] investigated the computing offloading in the energy harvesting devices, and devised an algorithm to reach the minimal execution cost. Lyu et al. [18] studied the cooperation offloading of several selfish devices, and proposed a distributed algorithm with the low time complexity to optimize the energy consumption. Kwak et al. [19] focused on computing offloading for various applications in single-user MEC system, and designed a dynamic algorithm which could satisfy the delay constraints and minimize the energy consumption. Jiang et al. [20] investigated the scheduling scheme for one multi-core mobile device, and designed an algorithm to optimize the energy consumption while stabilizing the queue. These works mainly focused on single-user system. It is more challengeable to design offloading strategy for multi-user system. Furthermore, as the IoT devices' number increases rapidly, a low-complexity algorithm is critical for the task offloading in multi-user system.

Lyu et al. [21] focused on the task scheduling problem to maximize the system utility, and designed an optimal scheme which could decrease the feedbacks from IoT devices. Mao et al. [22] focused on the resource management in multi-user MEC system, and built one optimization model to optimize the power while stabilizing the queue. However, these works gave few insights on the impact of channel condition on the devices' energy consumption. To tackle the above issues, we focus on the task offloading in multiple users MEC system, and propose a stochastic energy efficient optimization framework. And an effective algorithm which requires no prior knowledge about the task arrival or channel state's statistical information is designed to solve this optimization problem.

## 3 SYSTEM MODEL AND PROBLEM FORMULATION

### 3.1 System Model

One MEC system with a base station (BS) is considered. The BS has a MEC server providing services to  $n$  IoT devices in proximity. IoT devices can access the MEC server through the wireless channels, and offload the computation tasks for processing. By computation task offloading, the IoT devices can obtain better service and prolong the battery life. The set of the IoT devices is collected by  $I = \{1, 2, \dots, n\}$ , and a time-slotted system is considered, which is indexed by  $t \in \{0, 1, \dots, T - 1\}$  with slot length  $\tau$ . Table 1 gives the main notations.

TABLE 1  
NOTATIONS AND DEFINITIONS

Notion	Definition
$I$	IoT devices set
$\tau$	Time slot length
$P_i$	Transmit power of the IoT device $i$
$S(t)$	Number of available sub-channels in slot $t$
$h_i(t)$	Channel power gain for IoT device $i$ on the sub-channel in time slot $t$
$B$	Bandwidth of each sub-channel
$N_0$	Noise power spectral density
$R_i(t)$	Task offloading rate of IoT device $i$ in slot $t$
$\kappa_i(t)$	Offloading duration of IoT device $i$ during slot $t$
$e(t)$	Transmission energy consumption of all IoT devices in slot $t$
$A_i(t)$	Amount of computation tasks arrived at IoT device $i$ in slot $t$
$Q_i(t)$	Queue backlog of the IoT device $i$ in slot $t$
$D_i(t)$	Amount of computation tasks at IoT device $i$ which will be offloaded in slot $t$

### 3.2 Task Model and Offloading Model

For each IoT device  $i$ , let  $A_i(t)$  (in bits) represent the amount of arrived computation tasks. For generality, these IoT devices are considered heterogeneous. Therefore,  $A_i(t)$  can be different among different IoT devices. Note that we do not need any prior statistic information about  $A_i(t)$ , which is also hard to be obtained in real-life systems. This makes our model more extensive and applicable.

Let  $S(t)$  denote the number of available uplink sub-channels. For generality, we consider  $S(t)$  can be dynamic over different time slots. For each IoT device  $i$ , define  $P_i$  as its transmit power, and  $h_i(t)$  as the channel power gain in slot  $t$  [23]. Then, the achievable task offloading rate (in bit/s) is denoted by  $R_i(t)$  as follows,

$$R_i(t) = B \log_2 \left( 1 + \frac{P_i h_i(t)}{BN_0} \right), \quad (1)$$

where  $B$  is the sub-channel's bandwidth, and  $N_0$  is the noise power spectral density.

This article studies the task offloading for the IoT devices in the MEC system. Define the task offloading decisions as  $\kappa(t) = \{\kappa_1(t), \dots, \kappa_n(t)\}$ , where  $\kappa_i(t)$  denotes the offloading duration for IoT device  $i$ . Then, the available amount of computation tasks offloaded by device  $i$  is  $D_i(t) = R_i \kappa_i(t)$ .

Each IoT device maintains a task buffer to store the arrived but not yet offloaded tasks. Let  $Q_i(t)$  denote the task buffer's queue backlog at IoT device  $i$ . Since each IoT device can not offload more than what it has, it should be satisfied that

$$\kappa_i(t) \leq \frac{Q_i(t)}{R_i(t)}, \forall i \in I. \quad (2)$$

Consider each IoT device operates in narrow-band, thus it can only access a sub-channel at the same time [24]. Hence, it holds that

$$0 \leq \kappa_i(t) \leq \tau, \forall i \in I. \quad (3)$$

Together with (2) and (3), we can obtain that

$$0 \leq \kappa_i(t) \leq T_i(t), \forall i \in I, \quad (4)$$

where  $T_i(t) = \min\{Q_i(t)/R_i(t), \tau\}$ .

Similar to [21], we consider that each sub-channel can be accessed by the time division multiple access (TDMA), where a sub-channel can be accessed by different devices at different times during one slot. Moreover, a IoT device can access different sub-channels at different times to offload computation tasks during one slot [24]. Therefore, the total offloading duration of all the IoT devices must not exceed the time slot length of all the available sub-channels, as described in (5).

$$\sum_{i=1}^n \kappa_i(t) \leq S(t)\tau. \quad (5)$$

### 3.3 Task Queuing Model

Recall that  $Q_i(t)$  denotes the queue backlog of IoT device  $i$ , and  $D_i(t)$  represents the amount of offloaded computation tasks in device  $i$ . Then, the queue backlog in the next slot,  $Q_i(t+1)$ , is,

$$Q_i(t+1) = \max\{Q_i(t) - D_i(t), 0\} + A_i(t). \quad (6)$$

To reduce the queuing delay and maintain the queue stability of these IoT devices, for each IoT device  $i$ , we bound its average queue backlog  $q_i$  across the time slots, which is,

$$q_i = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{Q_i(t)\} < \varepsilon, \exists \varepsilon \in \mathbb{R}^+. \quad (7)$$

### 3.4 Energy Model

For each IoT device  $i \in I$ , the energy consumption for transmission hinges on the transmit power and offloading duration, i.e.,  $e_i(t) = P_i \kappa_i(t)$ . Thus, the energy consumed by all the IoT devices for transmission is,

$$e(t) = \sum_{i=1}^n P_i \kappa_i(t). \quad (8)$$

As the wireless channels' qualities are dynamic and change over different time slots, the task offloading speed and energy consumption for transmission during one slot also vary across time. Thus, we target at the average transmission energy consumption in long time scale which is expressed in (9).

$$e = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\}. \quad (9)$$

### 3.5 Optimization Problem

When the wireless channels are under better condition, the transmission speed could be larger. Thus, it is preferred to transmitting offloadable tasks when the wireless channels are better so as to reduce the overall transmission duration and the energy consumption. However, if we do not offload the tasks until the wireless channels become good enough, the queue backlog of the IoT devices might become very large and the IoT devices might be unstable. Therefore, we can see that tradeoff exists between the transmission energy consumption and queue backlog. In this article, a unified optimization problem is formulated for the computation offloading in IoT devices, which aims at minimizing average transmission energy consumption while guaranteeing the average queueing latency.

$$\min_{\kappa(t)} e = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \quad (10)$$

s.t. (4), (5) and (7).

**Remark:** Problem (10) is a stochastic optimization problem as the computation task arrivals and the wireless channel conditions vary across time randomly. Since the statistical information is generally hard to be acquired or predicted accurately in real systems, it is of great challenge to solve Problem (10) offline. Moreover, as the IoT devices' number rises rapidly, the solution space is increasing dramatically. How to solve the problem efficiently with low complexity is also very challenging. To attack the above challenges, an online task offloading algorithm is proposed which has no requirements on the statistical information of computation task arrivals or channel conditions, in Sec 4.

## 4 ENERGY EFFICIENT DYNAMIC OFFLOADING ALGORITHM DESIGN

In this section, taking advantage of Lyapunov optimization techniques [25], an energy efficient dynamic offloading algorithm called EEDOA is designed to solve Problem (10). By dynamically making the offloading decisions, EEDOA can achieve the tradeoff between the transmission energy consumption and queue backlog, and arbitrarily approximate the optimal transmission energy consumption while stabilizing the IoT devices.

### 4.1 Problem Transformation

We define  $\Theta(t)$  as the queue backlog matrix of the IoT devices, i.e.,  $\Theta(t) = (Q_i(t))$ . Let  $L(\Theta(t))$  denote the Lyapunov function, which is,

$$L(\Theta(t)) = \frac{1}{2} \sum_{i=1}^n Q_i^2(t). \quad (11)$$

In (11),  $L(\Theta(t))$  indicates the IoT devices' queue backlog state. A large  $L(\Theta(t))$  implies that at least a IoT device's queue backlog is large. If and only if the queue backlog of each IoT device is small,  $L(\Theta(t))$  would be small. Thus, we seek to reduce the value of  $L(\Theta(t))$  to maintain a low congestion state of the IoT devices. Define *conditional Lyapunov drift* as  $\Delta(\Theta(t))$  in (12).

$$\Delta(\Theta(t)) = \mathbf{E}\{L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)\}. \quad (12)$$

Recall that this article minimizes the transmission energy consumption and guarantees the IoT devices' queue backlog. Following Lyapunov optimization theory, combining transmission energy consumption with queue backlog, the *drift plus energy* is,

$$\Delta(\Theta(t)) + V \mathbf{E}\{e(t) | \Theta(t)\}. \quad (13)$$

In (13),  $V \geq 0$  is a tradeoff parameter between the transmission energy consumption and queue backlog. Particularly, a large  $V$  indicates that more weight is put on transmission energy consumption.

Next, the *drift plus energy's* upper bound is given by Theorem 1.

**THEOREM 1:** In slot  $t$ , if  $A_i(t)$  and  $R_i(t)$  are upper bounded by  $A_i^{max}$  and  $R_i^{max}$  over the time slots, the *drift plus energy* with any task offloading algorithm satisfies,

$$\begin{aligned} \Delta(\Theta(t)) + V \mathbf{E}\{e(t) | \Theta(t)\} &\leq C \\ &+ \sum_{i \in I} Q_i(t) \mathbf{E}\{A_i(t) - R_i(t) \kappa_i(t) | \Theta(t)\} \\ &+ V \sum_{i \in I} \mathbf{E}\{P_i \kappa_i(t) | \Theta(t)\}, \end{aligned} \quad (14)$$

where  $C = \frac{1}{2} [\sum_{i \in I} (A_i^{max})^2 + \sum_{i \in I} (R_i^{max} \tau)^2]$  is a constant.

*Proof:* Taking square on (6) and exploiting  $(\max[a - b, 0])^2 \leq a^2 + b^2 - 2ab$  for any  $a, b \geq 0$ , we can obtain (15).

$$\begin{aligned} Q_i^2(t+1) &\leq Q_i^2(t) + D_i^2(t) + A_i(t)^2 - 2Q_i(t)D_i(t) \\ &\quad + 2A_i(t) \max[Q_i(t) - D_i(t), 0]. \end{aligned} \quad (15)$$

Let  $\bar{D}_i(t)$  represent the actual amount of computation tasks offloaded from IoT device  $i$ . Thus, (16) holds.

$$\bar{D}_i(t) = \begin{cases} Q_i(t), & Q_i(t) \leq D_i(t) \\ D_i(t), & otherwise. \end{cases} \quad (16)$$

So it holds that  $\max[Q_i(t) - D_i(t), 0] = Q_i(t) - \bar{D}_i(t)$ . And we can rewrite (15) as,

$$\begin{aligned} Q_i^2(t+1) &\leq Q_i^2(t) + D_i^2(t) + A_i^2(t) \\ &\quad + 2Q_i(t)[A_i(t) - D_i(t)] - 2\bar{D}_i(t)A_i(t). \end{aligned} \quad (17)$$

Since  $\bar{D}_i(t)$  and  $A_i(t)$  are non-negative, it holds (18).

$$\begin{aligned} \frac{1}{2}[Q_i^2(t+1) - Q_i^2(t)] &\leq \frac{1}{2}[A_i^2(t) + D_i^2(t)] \\ &\quad + Q_i(t)[A_i(t) - D_i(t)]. \end{aligned} \quad (18)$$

Summing over all the devices on (18) and taking conditional expectations, (19) holds.

$$\begin{aligned} \Delta(\Theta(t)) &\leq \frac{1}{2} \sum_{i \in I} \mathbf{E}\{A_i^2(t) + D_i^2(t)|\Theta(t)\} \\ &\quad + \sum_{i \in I} Q_i(t) \mathbf{E}\{A_i(t) - D_i(t)|\Theta(t)\}. \end{aligned} \quad (19)$$

Since for any  $i \in I$ , there holds that  $R_i(t) \leq R_i^{max}$  and  $\kappa_i(t) \leq \tau$ , we obtain

$$D_i(t) = R_i \kappa_i(t) \leq R_i^{max} \tau. \quad (20)$$

According to (20) and  $A_i(t) \leq A_i^{max}$ , (21) can be obtained.

$$\sum_{i \in I} \mathbf{E}\{A_i^2(t) + D_i^2(t)|\Theta(t)\} \leq \sum_{i \in I} [(A_i^{max})^2 + (R_i^{max} \tau)^2]. \quad (21)$$

Let  $C$  equal to  $\frac{1}{2} \sum_{i \in I} [(A_i^{max})^2 + (R_i^{max} \tau)^2]$ , and add  $V \mathbf{E}\{e(t)|\Theta(t)\}$  to (19). Thus,

$$\begin{aligned} \Delta(\Theta(t)) + V \mathbf{E}\{e(t)|\Theta(t)\} &\leq C + V \mathbf{E}\{e(t)|\Theta(t)\} \\ &\quad + \sum_{i \in I} Q_i(t) \mathbf{E}\{A_i(t) - D_i(t)|\Theta(t)\}. \end{aligned} \quad (22)$$

Substituting (8) into (22), it yields (14).  $\square$

## 4.2 Energy Efficient Dynamic Offloading Algorithm

In this subsection, we minimize the *drift plus energy's* upper bound, and devise an optimal algorithm, EE-DOA. By minimizing the upper bound in each time slot, EEDOA can effectively reduce the average energy consumption for transmission and maintain the queue backlog of each IoT device at a low level.

In each time slot, EEDOA makes the offloading decisions  $\kappa(t)$  to minimize the the *drift plus energy's* upper bound, expressed by,

$$\begin{aligned} \min_{\kappa(t)} \{C + \sum_{i \in I} Q_i(t)[A_i(t) - R_i(t)\kappa_i(t)] + V \sum_{i \in I} P_i \kappa_i(t)\}. \\ \text{s.t. (4), (5)}. \end{aligned} \quad (23)$$

Since  $C$  and  $A_i(t)$  are constant in each slot  $t$ , (23) can be reduced to (24).

$$\min_{\kappa(t)} \sum_{i \in I} [V P_i - Q_i(t) R_i(t)] \kappa_i(t). \quad (24)$$

Transforming the above minimization problem to the maximization problem, we have (25).

$$\begin{aligned} \max_{\kappa(t)} \sum_{i \in I} \omega_i(t) \kappa_i(t) \\ \text{s.t. (4), (5)}, \end{aligned} \quad (25)$$

where  $\omega_i(t) = Q_i(t) R_i(t) - V P_i$ .

Problem (25) can be regarded as the linear relaxation of a knapsack problem.  $S(t)\tau$  is the capacity of the knapsack, and  $\omega_i(t)$  can be considered as the item's unit value. Hence, for this linear relaxation

### Algorithm 1 Energy Efficient Dynamic Offloading Algorithm (EEDOA)

- 1: **for all**  $i \in I$  **do**
- 2:   Calculate the  $R_i(t)$ ,  $T_i(t)$  and  $\omega_i(t)$ .
- 3: **end for**
- 4: Sort all the devices  $i$  in the descending order of  $\omega_i(t)$ .
- 5: Set the index  $\chi$  according to (26).
- 6: **for all**  $i \in I$  **do**
- 7:   Set the offloading decision  $\kappa_i(t)$  according to (29).
- 8: **end for**

knapsack problem, the optimal solution is to select the *item* with highest non-negative  $\omega_i(t)$  to fulfill the knapsack [26]. Therefore, we sort the IoT devices in the descending order of  $\omega_i(t)$ , where  $\omega_i(t) \geq \omega_{i+1}(t)$ . The devices with larger values of  $\omega_i(t)$  have higher offloading priorities, and are filled into the knapsack prior to other devices. Then, the interruption condition of the knapsack filling process is that the residual capacity of the knapsack is empty or the unit value of the selected *item* is negative. Therefore, the index of interruption *item*  $\chi$  can be derived as follows,

$$\chi = \min\{\chi_1, \chi_2\}, \quad (26)$$

where

$$\chi_1 = \arg \min_i \sum_{j=1}^i T_j(t) > S(t)\tau, \quad (27)$$

$$\chi_2 = \arg \max_i \omega_i(t) \geq 0. \quad (28)$$

Thus, we can obtain the optimal offloading decisions  $\kappa^*(t)$ ,

$$\kappa_i^*(t) = \begin{cases} T_i(t), & i < \chi \\ \min\{S(t)\tau - \sum_{i=1}^{\chi-1} T_i(t), T_\chi(t)\}, & i = \chi \\ 0, & i > \chi. \end{cases} \quad (29)$$

**Remark:** Recall that there exists tradeoff between transmission energy efficiency and queue backlog. For each IoT device  $i$ , combining its transmission energy consumption and queue length, EEDOA defines  $\omega_i(t)$  as its *unit offloading profit*.  $\omega_i(t)$  hinges on the current queue backlog and channel condition, as well as varies with time. In each time slot  $t$ , by selecting the IoT devices with higher non-negative  $\omega_i(t)$  to offload tasks, EEDOA maximizes the overall *offloading profit* for all the IoT devices. As a result, EEDOA can reduce both transmission energy consumption and queue length effectively. In addition, according to the definition of  $\omega_i(t)$ , EEDOA can achieve the tradeoff between queue backlog and transmission energy consumption arbitrarily by adjusting  $V$ .

Algorithm 1 presents the details of the EEDOA algorithm.

## 5 ALGORITHM ANALYSIS FOR EEDOA

We first analyze the EEDOA's performance through mathematical analysis, then give the time complexity of EEDOA.

Define  $\bar{Q}$  as the time-average queue backlog of the IoT devices, expressed in (30).

$$\bar{Q} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^n \mathbf{E}\{Q_i(t)\}. \quad (30)$$

To derive the upper bounds of  $e$  and  $\bar{Q}$ , we present that there exists an optimal strategy which is independent of the queue length, and makes the offloading decisions  $\kappa(t)$  following some fixed probability distribution, and can optimize the transmission energy efficiency. The details are given in Lemma 1.

*LEMMA 1:* For any task arrival rate  $\lambda$  satisfying  $\lambda \in \Lambda$ , a task offloading strategy  $\pi^*$  exists, which does not depend on the current queue length and satisfies,

$$\begin{aligned} \mathbf{E}\{e^{\pi^*}(t)\} &= e^*(\lambda); \\ \mathbf{E}\{A_i(t)\} &\leq \mathbf{E}\{R_i(t)\kappa_i^{\pi^*}(t)\}, \end{aligned}$$

where  $\Lambda$  represents the system capacity, and  $e^*(\lambda)$  represents the optimal transmission energy consumption with  $\lambda$ .

*Proof:* Caratheodory's theorem can be used to prove Lemma 1 [25]. For the sake of readability, the detailed proofs are omitted here.  $\square$

Recall that the task arrival rate  $A_i(t)$  of each IoT device is upper bounded by  $A_i^{max}$ . Thus, the energy consumption for transmission would be upper bounded by a finite value  $\hat{e}$  and lower bounded by a finite value  $\check{e}$ . Then, by applying Lemma 1, Theorem 2 gives the average transmission energy consumption's upper bound and queue length's upper bound.

*THEOREM 2:* Suppose there exists a positive  $\epsilon$  satisfying  $\lambda + \epsilon \in \Lambda$ . For the given  $V$ , the transmission energy consumption of EEDOA satisfies,

$$e^{EEDOA} \leq e^* + \frac{C}{V}. \quad (31)$$

In addition, the average queue length of EEDOA is also upper bounded in (32).

$$\bar{Q} \leq \frac{C + V(\hat{e} - \check{e})}{\epsilon}, \quad (32)$$

where  $C$  is a constant given by Theorem 1, and  $e^*$  represents the optimal transmission energy consumption with  $\lambda$ .

*Proof:* According to Lemma 1, for the task arrival rate  $\lambda + \epsilon$ , a randomized policy  $\pi'$  exists, which satisfies,

$$\mathbf{E}\{e^{\pi'}(t)\} = e^*(\lambda + \epsilon); \quad (33)$$

$$\mathbf{E}\{A_i(t)\} + \epsilon \leq \mathbf{E}\{R_i(t)\kappa_i^{\pi'}(t)\}. \quad (34)$$

Since EEDOA minimizes (22)'s R.H.S., for the offloading strategy  $\pi'$ , we can obtain that

$$\begin{aligned} \Delta(\Theta(t)) + V\mathbf{E}\{e(t)|\Theta(t)\} &\leq C + V\mathbf{E}\{e^{\pi'}(t)|\Theta(t)\} \\ &+ \sum_{i \in I} Q_i(t)\mathbf{E}\{A_i(t) - R_i(t)\kappa_i^{\pi'}(t)|\Theta(t)\}. \end{aligned} \quad (35)$$

Plugging (33) and (34) into (35), it can be obtained that

$$\Delta(\Theta(t)) + V\mathbf{E}\{e(t)|\Theta(t)\} \leq C + Ve^*(\lambda + \epsilon) - \epsilon \sum_{i \in I} Q_i(t). \quad (36)$$

Taking expectations on (36), and using iterated expectations, we can obtain

$$\begin{aligned} \mathbf{E}\{L(\Theta(t+1)) - L(\Theta(t))\} + V\mathbf{E}\{e(t)\} \\ \leq C + Ve^*(\lambda + \epsilon) - \epsilon \sum_{i \in I} \mathbf{E}\{Q_i(t)\}. \end{aligned} \quad (37)$$

Summing (37) over the slots, it holds,

$$\begin{aligned} \mathbf{E}\{L(\Theta(T))\} - \mathbf{E}\{L(\Theta(0))\} + V \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \\ \leq CT + VT e^*(\lambda + \epsilon) - \epsilon \sum_{t=0}^{T-1} \sum_{i \in I} \mathbf{E}\{Q_i(t)\}. \end{aligned} \quad (38)$$

For generality, consider that the queue lengths of the IoT devices are empty when  $t = 0$ . Thus, we can obtain  $L(\Theta(0)) = 0$ . Moreover, since it holds that  $L(\Theta(T)) \geq 0$ , it holds,

$$V \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \leq CT + VT e^*(\lambda + \epsilon) - \epsilon \sum_{t=0}^{T-1} \sum_{i \in I} \mathbf{E}\{Q_i(t)\}. \quad (39)$$

Since  $Q_i(t)$  and  $\epsilon$  are non-negative, we obtain (40).

$$V \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \leq CT + VT e^*(\lambda + \epsilon). \quad (40)$$

By dividing (40) by  $VT$ , it yields

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \leq \frac{C}{V} + e^*(\lambda + \epsilon). \quad (41)$$

Let  $\epsilon \rightarrow 0$ ,  $T \rightarrow \infty$  and apply the Lebesgues dominated convergence theorem, we obtain (31).

According to (39), we also obtain

$$\begin{aligned} \epsilon \sum_{t=0}^{T-1} \sum_{i \in I} \mathbf{E}\{Q_i(t)\} \\ \leq CT + VT e^*(\lambda + \epsilon) - V \sum_{t=0}^{T-1} \mathbf{E}\{e(t)\} \\ \leq CT + VT(\hat{e} - \check{e}) \end{aligned} \quad (42)$$

By dividing (42) by  $\epsilon T$ , and letting  $T \rightarrow \infty$ , we obtain (32).  $\square$

**Remark:** (31) shows that transmission energy consumption's upper bound decreases as  $V$  rises. However, (32) shows that the queue length's upper bound rises as  $V$  increases. Nevertheless, (7) is satisfied if  $\varepsilon$  is set as  $\frac{C+V(\hat{\varepsilon}-\varepsilon)}{\varepsilon}$ . We use  $O(1/V)$  to represent the upper bound of gap of energy consumption, and  $O(V)$  to represent the gap of queue length [21]. Combing (31) and (32), we can see that EEDOA achieves an  $[O(1/V), O(V)]$  tradeoff between transmission energy efficiency and experience. By changing  $V$ , EEDOA can realize different balances between energy consumption and queue backlog. Besides, Theorem 2 also indicates that the minimal transmission energy consumption can be approximated by EEDOA while guaranteeing that all the devices are stable.

Next, we provide the time complexity analysis for EEDOA. According to Algorithm 1, for the two loops (line 1-3 and line 6-8), EEDOA traverses all the IoT devices once. Thus, each loop terminates in  $O(n)$  operations, where  $n$  is the IoT devices' number. For line 4, the process of sorting devices takes  $O(n \log n)$  operations with the quicksort algorithm. For line 7, it takes  $O(n)$  operations in the worst case. Thus, the EEDOA's time complexity is  $O(n \log n)$ .

## 6 EVALUATION

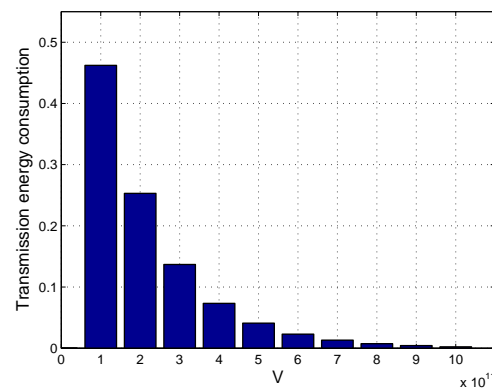
This section demonstrates that EEDOA can adapt to various parameter changes. We also provide the comparison experiments to show the EEDOA's effectiveness.

In the experiments, we consider 100 IoT devices offloading computation tasks. The slot length  $\tau = 1$  s. The offloadable task arriving rate and the channel state are set to follow certain fixed statistical distributions. The amount of data arriving at IoT device  $i$  per second is set to be uniform distributed within  $[0, 2200]$  bits, i.e.,  $A_i(t) \sim U[0, 2200]$  bits. For the wireless channel, a small-scale Rayleigh fading model is adopted, and  $h_i(t)$  is an exponential distribution with the unit mean, i.e.,  $h_i(t) \sim E(1)$  [22]. We set the number of available sub-channels as  $S(t) \sim U[10, 30]$ , and the transmit power of each IoT device as  $P_i \sim U[10, 200]$  mW [21]. Beside,  $B$  is set as 1 MHz and  $N_0$  is set as  $10^{-6}$  W/Hz. To improve the reliability of the experiments, we run 3000 times for every setting, and average the results. Note that although in the experiments, the offloadable task arrivals and channel states are set according to fixed statistical distributions, EEDOA actually requires no statistics for the information in advance.

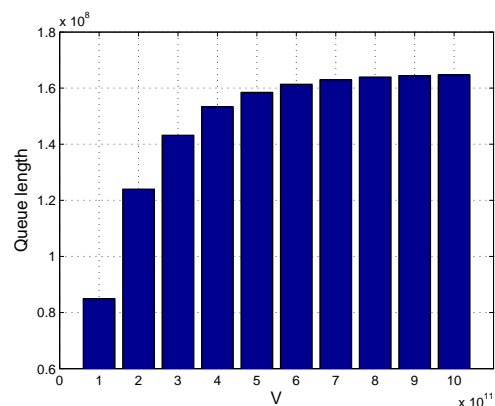
### 6.1 Parameter Analysis

#### 6.1.1 Effect of tradeoff parameter

In Fig. 1, we plot the transmission energy consumption and queue length with different  $V$ . Fig. 1(a) shows the relationship between the transmission energy consumption and tradeoff parameter  $V$ . We can see



(a) Transmission energy consumption



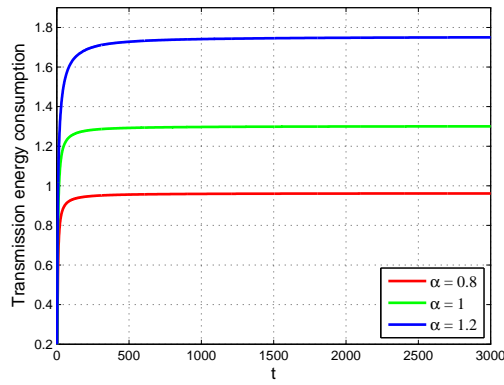
(b) Queue length

Fig. 1. Transmission energy consumption and queue length with different values of  $V$

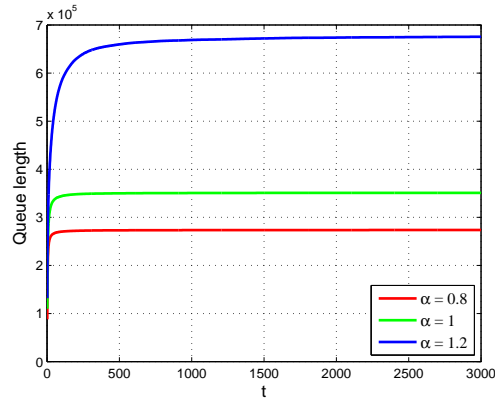
that as  $V$  rises, the transmission energy consumption decreases, which conforms (31) in Theorem 2. The reason is that a larger  $V$  means putting more weight on transmission energy consumption, and EEDOA would dynamically adjust the offloading decisions to reduce the transmission energy consumption. However, Fig. 1(b) shows that when  $V$  increases, the queue length also rises. This phenomena conforms (32) in Theorem 2. Nevertheless, the queue length would converge gradually with more increase of  $V$ , demonstrating that EEDOA's queue length would be bounded. From Fig. 1, it can be observed that by adjusting  $V$ , EEDOA can balance the transmission energy consumption and queue length. Besides, by increasing the value of  $V$  sufficiently large, EEDOA can approach the optimal energy consumption and stabilize all the IoT devices.

#### 6.1.2 Effect of arrival rate

In Fig. 2, we plot the transmission energy consumption and queue length with different task arrival rates. The arrival rate for each IoT device is set as  $\alpha \cdot A_i(t)$ , where  $\alpha = 0.8, 1$  and  $1.2$ , respectively. Fig. 2(a) illustrates that the transmission energy consumption rises as the arrival rate rises. It is because as the arrival rate



(a) Transmission energy consumption



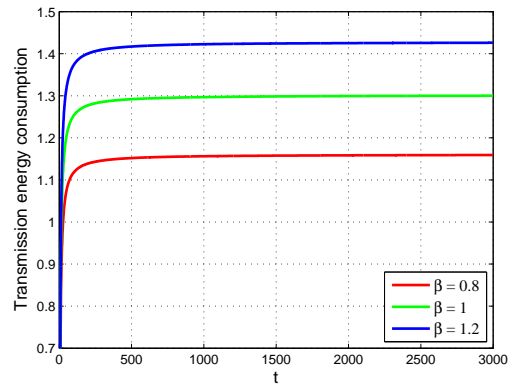
(b) Queue length

Fig. 2. Transmission energy consumption and queue length with different arrival rates

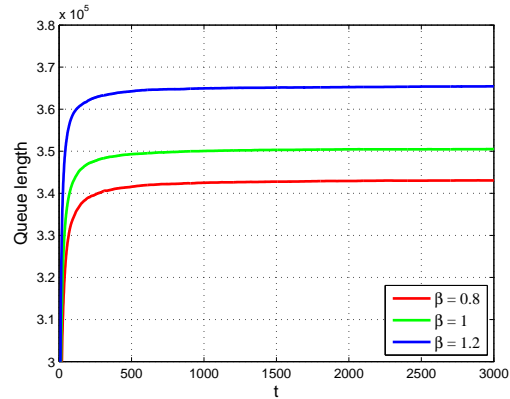
rises, the amount of computation tasks also increases. As a result, it would consume more energy to offload the computation tasks. Similarly, Fig. 2(b) shows that the queue length rises as arrival rate rises. In addition, we can see that with different arrival rates, both the transmission energy consumption and queue length of EEDOAs would converge quickly. This shows that EEDOAs can dynamically tune the offloading decisions to adjust to the change of arrival rate, and make the IoT devices stable quickly.

### 6.1.3 Effect of transmit power

In Fig. 3, we plot the transmission energy consumption and queue length with different  $P_i$ . For each IoT device, we set the transmit power as  $\beta \cdot P_i$ , where  $\beta = 0.8, 1$  and  $1.2$ , respectively. Fig. 3(a) shows that as the transmit power rises, the energy consumption rises. It is because that with the rise of transmit power, the energy consumption of each IoT device for transmitting data would also increase. Consequently, the overall transmission energy consumption of EEDOAs would rise. Fig. 3(b) shows that the queue length rises with the increase of transmit power. The reason is that when the transmit power increases, EEDOAs would reduce the amount of computation tasks offloaded to



(a) Transmission energy consumption



(b) Queue length

Fig. 3. Transmission energy consumption and queue length with different transmit powers

decrease the transmission energy consumption, which results in the increase of queue length.

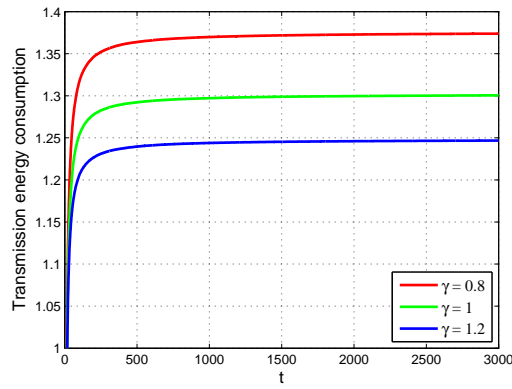
### 6.1.4 Effect of channel power gain

Fig. 4 plots the transmission energy consumption and queue length with different channel power gains. For each IoT device, we set the channel power gain to be the exponential distribution with the mean of  $\gamma \cdot \mu$ , where  $\gamma = 0.8, 1$  and  $1.2$ , respectively. Fig. 4(a) shows that as channel power gain rises, the transmission energy consumption reduces. With the rise of channel power gain, the offloading rate would increase, and thus, the transmission duration for offloading tasks decreases. Consequently, the transmission energy consumption of EEDOAs would also decrease. We can see the similar phenomena in Fig. 4(b) that the queue length decreases as channel power gain rises. The reason is that as the channel power gain and offloading rate rise, the tasks offloaded would rise; therefore, the queue length of EEDOAs would decrease.

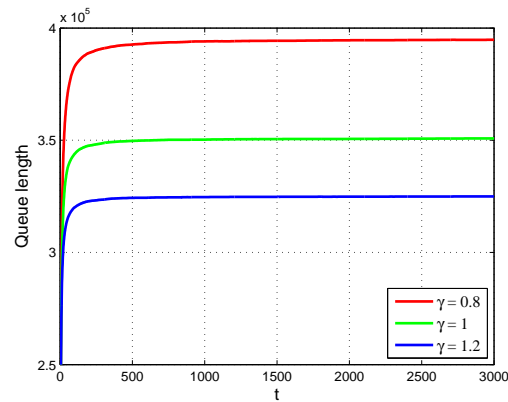
### 6.1.5 Effect of number of IoT devices

Fig. 5 plots the transmission energy consumption and queue length with different numbers of IoT devices.





(a) Transmission energy consumption



(b) Queue length

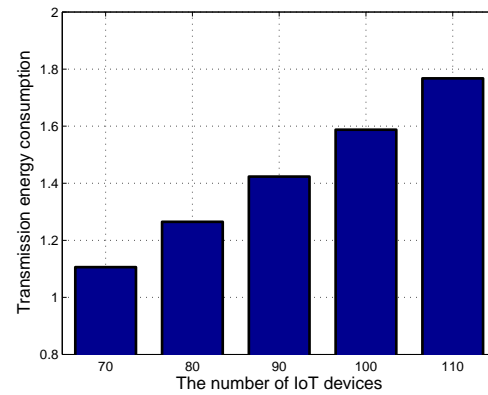
Fig. 4. Transmission energy consumption and queue length with different channel power gains

The IoT devices' number ranges from 70 to 110 with an increment of 10. From Fig. 5(a), we can see that as IoT devices' number rises, the transmission energy consumption rises. It is because that when the IoT devices' number rises, the amount of offloaded computation tasks also rises, which causes the rise of the transmission energy consumption. Fig. 5(b) shows that as the IoT devices' number rises, the total queue length increases. The reason is that the amount of offloaded computation tasks would rise as the IoT devices' number rises. Only part of the increased computation tasks are offloaded, and the rest are stored in the task buffer waiting for transmitting, which leads to the increase of queue length.

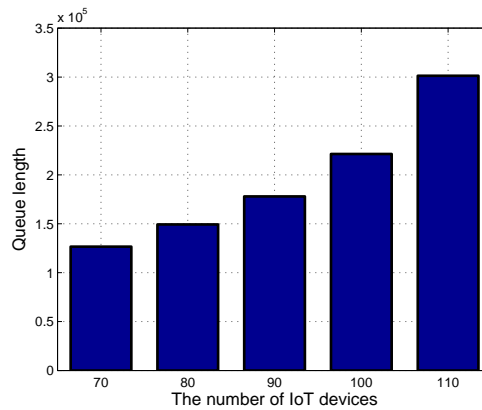
## 6.2 Comparison Experiments

To further evaluate the EEDOA's performance, we compare EEDOA with two baseline algorithms:

- **Equal allocation strategy:** In slot  $t$ , the offloading duration is allocated among all the IoT devices equally.
- **Queue-weighted strategy:** In slot  $t$ , the offloading duration is allocated among the IoT devices



(a) Transmission energy consumption

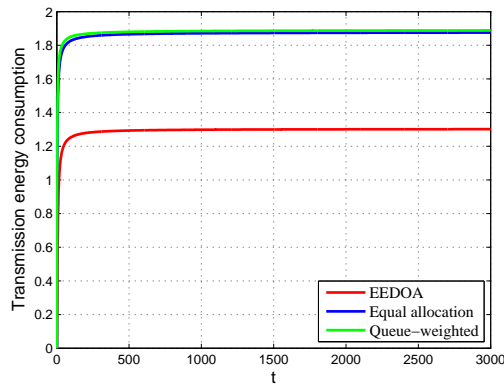


(b) Queue length

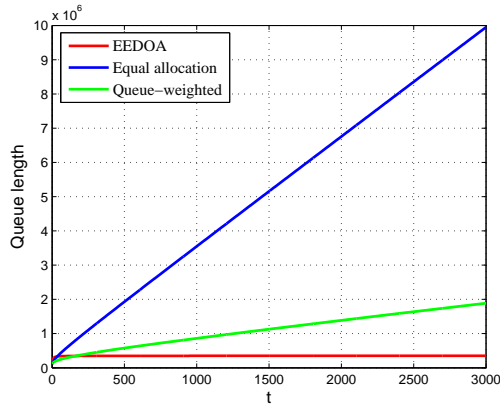
Fig. 5. Transmission energy consumption and queue length with different numbers of IoT devices

according to their weighted queue length. Particularly, IoT devices with larger queue length have higher priorities, and the offloading durations allocated to them are longer.

In Fig. 6, we plot the transmission energy consumption and queue length with different algorithms. Both the transmission energy consumption and queue length of our EEDOA are the lowest. It is demonstrated that EEDOA can decrease the transmission energy consumption as well as the queue length effectively. It is because EEDOA can dynamically make the offloading decisions among the IoT devices to adapt to the channel dynamics and queue length. In addition, we can see that the transmission energy consumption of Equal allocation strategy and Queue-weighted strategy is close; however, the queue length of Queue-weighted strategy is less than the Equal allocation strategy. The reason is that Queue-weighted strategy allocates longer offloading durations to the IoT devices with larger queue length, reducing the overall queue length of the IoT devices. Nevertheless, the Queue-weight strategy does not take into account the channel dynamics. Our EEDOA considers both the dynamic channel states and the queue length



(a) Transmission energy consumption



(b) Queue length

Fig. 6. Transmission energy consumption and queue length with three different algorithms

information. Fig. 6 shows the superiority of EEDO in reducing the transmission energy consumption as well as the queue length effectively.

## 7 CONCLUSION

In this article, an online and polynomial-time-complexity algorithm EEDO has been proposed for offloading in MEC to approximate the minimal energy consumption and guarantee the IoT devices' delay performance. It needs no prior statistic knowledge related with task arrival or channel condition. EEDO can also make arbitrary tradeoff between transmission energy efficiency and queue backlog. A close-to-optimal transmission energy consumption can be acquired by EEDO while guaranteeing the queue backlog's upper bound. Experiment results show the EEDO's effectiveness in reducing the transmission energy consumption and keeping the queue backlog of IoT devices at a low level.

## REFERENCES

[1] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X. y. Li, "S2m: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 88–100, Feb 2017.

[2] H. Duan, Y. Zheng, C. Wang, and X. Yuan, "Treasure collection on foggy islands: Building secure network archives for internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[3] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, March 2017.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[5] X. Chen, Q. Shi, L. Yang, and J. Xu, "Thriftyedge: Resource-efficient edge computing for intelligent iot applications," *IEEE Network*, vol. 32, no. 1, pp. 61–65, Jan 2018.

[6] A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani, and C. Z. Patrikakis, "A cooperative fog approach for effective workload balancing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 36–45, March 2017.

[7] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.

[8] Y. Wu, J. Chen, L. P. Qian, J. Huang, and X. S. Shen, "Energy-aware cooperative traffic offloading via device-to-device cooperations: An analytical approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 97–114, Jan 2017.

[9] F. Liu, P. Shu, and J. C. S. Lui, "Appatp: An energy conserving adaptive mobile-cloud transmission protocol," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3051–3063, Nov 2015.

[10] Z. Sheng, C. Mahapatra, V. C. M. Leung, M. Chen, and P. K. Sahu, "Energy efficient cooperative computing in mobile wireless sensor networks," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 114–126, Jan 2018.

[11] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, Oct 2015.

[12] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in c-ran with mobile cloud," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2017.

[13] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

[14] Y. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and cpu time allocation for mobile edge computing," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[15] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, April 2017.

[16] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, May 2016.

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

[18] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Distributed online optimization of fog computing for selfish devices with out-of-date information," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7704–7717, Nov 2018.

[19] 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.

[20] Z. Jiang and S. Mao, "Energy delay trade-off in cloud offloading for multi-core mobile devices," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[21] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2606–2615, Nov 2017.

[22] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user

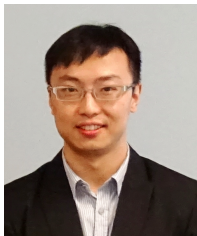
- mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5994–6009, Sept 2017.
- [23] Q. Liu, T. Han, N. Ansari, and G. Wu, "On designing energy-efficient heterogeneous cloud radio access networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 3, pp. 721–734, Sept 2018.
- [24] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5g era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, March 2016.
- [25] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
- [26] D. Pisinger, "Algorithms for knapsack problems," *North-Holland Mathematics Studies*, vol. 132, pp. 213–257, 1995.



**Wen Wu** received the B.E. degree from South China University of Technology, Guangzhou, China, in 2012 and the M.A.Sc. degree from University of Science and Technology of China, Hefei, China, in 2015. He is current a Ph.D. candidate in University of Waterloo.



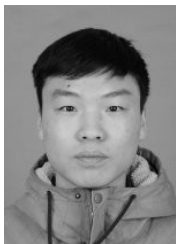
**Ying Chen** received the BEng degree from Beijing University of Posts and Telecommunications in 2012, and the PhD degree from Tsinghua University in 2017. She is an assistant professor in the Computer School at Beijing Information Science and Technology University.



**Ning Zhang** is an Assistant Professor at Texas A&M University-Corpus Christi, USA. He received the Ph.D degree from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively.



**Xuemin (Sherman) Shen** received Ph.D. degrees (1990) from Rutgers University, New Jersey (USA). Dr. Shen is a University Professor, in Department of Electrical and Computer Engineering, University of Waterloo, Canada. He serves as the Editor-in-Chief for IEEE Internet of Things Journal, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.



**Yongchao Zhang** received the BEng degree from Beijing Information Science and Technology University in 2017. He is currently a graduate student in Beijing Information Science and Technology University.



**Xin Chen** received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China. He is currently a Professor in Beijing Information Science and Technology University.