Research Article

Wireless-Uplinks-Based Energy-Efficient Scheduling in Mobile Cloud Computing

Xing Liu,¹ Chaowei Yuan,¹ Zhen Yang,² and Enda Peng¹

¹*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian, Beijing 100876, China*

²*School of Computer Science, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian, Beijing 100876, China*

Correspondence should be addressed to Xing Liu; liuxing_bupt@qq.com

Received 16 November 2014; Accepted 2 February 2015

Academic Editor: Bao Rong Chang

Copyright © 2015 Xing Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile cloud computing (MCC) combines cloud computing and mobile internet to improve the computational capabilities of resource-constrained mobile devices (MDs). In MCC, mobile users could not only improve the computational capability of MDs but also save operation consumption by offloading the mobile applications to the cloud. However, MCC faces the problem of energy efficiency because of time-varying channels when the offloading is being executed. In this paper, we address the issue of energy-efficient scheduling for wireless uplink in MCC. By introducing Lyapunov optimization, we first propose a scheduling algorithm that can dynamically choose channel to transmit data based on queue backlog and channel statistics. Then, we show that the proposed scheduling algorithm can make a tradeoff between queue backlog and energy consumption in a channel-aware MCC system. Simulation results show that the proposed scheduling algorithm can reduce the time average energy consumption for offloading compared to the existing algorithm.

1. Introduction

Mobile devices (MDs) are increasingly becoming an essential part of human life [1]. As the most effective and convenient tools, MDs are not bounded by time and place. However, the limited computational power, storage space, and battery lifetime of existing MDs significantly limit their ability to execute resource-intensive applications [2]. Fortunately, mobile cloud computing (MCC), which combines cloud computing with mobile internet, can provide service via Infrastructure as a Service (IaaS) platform [3]. Using this IaaS platform, on the one hand, the performance of MDs can be improved by offloading mobile applications to the cloud servers. On the other hand, as users offload the mobile applications to the cloud, the data, which are transmitted on both wired and wireless networks, are increased rapidly, the communication overhead of MD will significantly consume battery energy

[4]. Therefore, reducing the energy consumption of transmission is one of the most significant issues in MCC.

The issue about energy efficiency of data transmission in wireless networks was investigated in [5–7]. In [5], Zafer and Modiano used a novel continuous-time optimal-control formulation and Lagrangian duality, to propose an optimal transmission scheduling that can dynamically adapt to the rate over time considering the channel variations to minimize the transmission energy cost. In [6], Fu and van der Schaar investigated the structure-aware online learning for the energy-efficient and delay-sensitive transmission. In [7], Neely exploited time-varying channel conditions to design an energy-efficient control (EEC) algorithm based on the Lyapunov optimization. However, all the above papers only consider a single user transmitting data over a time-varying channel. To this end, much research work [8, 9] has focused on the issue of energy-efficiency for multiuser multichannel

scenario. In [8], Li and Neely consider a wireless base station serving N users through N time-varying channels and propose a dynamic channel acquisition algorithm (DCAA) that all mobile users dynamically share all wireless channels. In [9], Xiang et al. used discrete-time stochastic dynamic program, to propose an approximate dynamic programming (ADP) that can dynamically select channel and execute data transmission scheduling with time-varying channel conditions. The basic idea of the proposed schemes is optimized energy-efficiency of downlink data transmission between MDs and base stations.

Although all of the above work focused on the wireless downlinks, the energy efficiency of the wireless uplink is lacking. In fact, the energy consumption of uploading data is more than that of downloading data [10]; hence reducing the energy consumption of uploading data is the key to save the energy consumption on MDs in the MCC system. In [11], Ra et al. implemented Lyapunov optimization framework on MDs in multiple wireless uplinks environments and designed a stable and adaptive link selection algorithm (SALSA) with time-varying V values for meeting different applications' delay tolerances. This SALSA algorithm can automatically select channels and requires channel state information to decide whether and when to defer a transmission. However, the SALSA algorithm supports each user with a dedicated-channel when it has already selected a channel while an application is executing. In this way, user might miss better transmission opportunity on other channels; it is not so good for reducing MDs' energy consumption. For example, there are two users (1 and 2) uploading data over two time-varying channels (A and B). Let SALSA algorithm allocate channel A and channel B to user 1 and user 2 at the beginning of uploading, respectively. We assume user 1 has low service rate on channel A and high service rate on channel B, while user 2 has low service rate on channel B and high service rate on channel A after a period of time. According to the SALSA algorithm, on the one hand, users may defer the data transfer until their channels' service rate is high enough, rather than reallocate channel. On the other hand, as the queue backlog increases, users may transmit the data to cloud with the low service rate. Indeed, the energy-delay tradeoff can be provided. However, the high energy consumption is incurred, because the SALSA algorithm cannot reallocate channel and users may upload data with the low service rate when the queue backlog is high enough.

In this paper, based on the MCC system with IaaS platform, we address the issue of energy-efficient scheduling considering the resource-intensive applications, which need to upload large data to the cloud such as mHealth, mobile commerce, and mobile office. In our scheduling algorithm, mobile user chooses appropriate channels to transmit applications' data packet with the queue backlog and channel statistics every T time unit. In the above example, our scheduling algorithm can transmit user 1's data over channel B when the service rate of channel B is higher than that of channel A. Compared with the SALSA algorithm, our scheduling algorithm has a lower queue backlog because it can transmit user's data over any channel that has a high service rate support user. Thus, for the same parameter V , which control

the tradeoff between energy consumption and queue backlog, our scheduling algorithm has the lower energy consumption than SALSA algorithm. Our main contribution includes the following.

- (i) Adopting the framework of Lyapunov optimization, we propose a two-time dynamic offloading (T2DO) algorithm to decrease the energy consumption on MDs by considering queue backlog and channel states.
- (ii) We demonstrate that the proposed algorithm can approach the optimal energy consumption within $O(1/V)$ deviation, and a tradeoff in average queue backlog is $O(V)$.
- (iii) We compare the performance of our T2DO algorithm with SALSA and random and minimum-delay algorithms using simulation. The results show that, by appropriately choosing V , the T2DO algorithm outperforms the other three algorithms, with smaller time average energy consumption while achieving queue stability.

The remainder of this paper is organized as follows. In Section 2, the review of related work is presented. In Section 3, we present the system model and problem statement in the MCC. In Section 4, according to Lyapunov optimization, we propose a T2DO algorithm which makes a tradeoff between queue backlog and energy consumption. The performance analysis of T2DO algorithm is given in Section 5. In Section 6, we compare the performance of the T2DO algorithm with SALSA and random and minimum-delay algorithms using simulation. Section 7 concludes this paper and provides future directions.

2. Related Work

MCC is an emerging technology to extend the capabilities of MDs by offloading. According to [12], not all applications offloading can extend battery life of MDs. Thus, a vast amount of previous work [10, 13–16] has been investigating the scheme of applications offloading. In [10], Altamimi et al. proposed an energy model for the task offloading to the cloud. It minimized power consumption on the MDs under the constraints of the responsiveness and accuracy of demands of interactive perception tasks. In [13], Chen formulated the decision problem of computation offloading among mobile users as a decentralized computation offloading game and proposed a computation offloading mechanism based on game-theoretic approach to save energy on the MD. In [14], Yang et al. discussed an assisting execution offloading approach by reducing the size of transferred state and proposed an execution offloading scheme to improve the performance of the MCC significantly in terms of execution time and energy consumption. In [15], Zhang et al. investigated collaborative task execution between MD and cloud clone for mobile applications under the stochastic wireless channel. In [16], Zhang et al. proposed an energy-optimal execution strategy for the MCC under the stochastic wireless channel. For the mobile execution, it minimized the computation

energy by dynamically configuring the clock frequency of the chip. For the cloud execution, it minimized the transmission energy by optimally scheduling data transmission across the stochastic wireless channel.

With offloading, the data transfer between MD and cloud is increasing rapidly; thus the higher data transmission energy consumption is incurred, especially with a bad wireless channel. Some literatures have studied the energy issues of data transmission for offloading. In [17], Kumar and Lu considered a fixed computation scheduling with a fixed data rate model in the MD for the wireless channel. In [18], Huang et al. presented a dynamic offloading algorithm to transfer data to save energy on the MD while meeting the application execution time. In [19], Tilevich and Kwon presented a determined functionality to offload at runtime, which can maximize efficiency by automating program transformation. In [9], Xiang et al. presented a flexible link selection and data transmission scheduling and proposed a scalable approximate dynamic programming (ADP) algorithm based on the discrete-time stochastic dynamic program to reduce the average energy consumed for delivering a packet.

In a real-world application, uplink power consumption dominates the wireless power budget due to RF power requirements for reliable transmission over long distances; thus energy-efficiency of uploading data is a key issue in MCC due to energy-poverty of MDs. Some researchers have worked on the energy-optimal scheduling in wireless uplinks. In [20], Katanaras et al. considered clustered cooperation and investigated effective techniques for managing uplinks of intercluster interference to improve users' performance in terms of both spectral and energy efficiency. In [21], Miao et al. developed an energy-efficient scheme with significantly lower complexity when compared to iterative approaches in an uplink OFDMA system. This scheme allocates the system bandwidth among all users to optimize energy efficiency across the whole network. In [22], Liu et al. proposed a dynamic carrier aggregation (DCA) scheme to improve the energy efficiency of uplink communications. In [23], Deb and Monogioudis proposed LeAP, a measurement data-driven machine learning paradigm for power control to manage uplink interference in LTE. The authors in [11] derived the energy-efficient scheduling automatically select channels and requires channel state information to decide whether and when to defer data uploading. However, it cannot reselect channels while an application is executing, even though the service rate of other channels is higher than that of current channel in some time.

This paper investigates energy-efficient scheduling for wireless uplinks in the MCC. Compared to previous works, this paper has several differences. First, we employ the multiqueueing model and consider MD can reselect appropriate channel to transmit data while an application is executing. Second, we provide the theoretical framework of data uploading, in which MD first calculates the amount of data transfer of each offloading application to balance queue length among all queues; then MD chooses appropriate channels to transmit its data according to the queue backlog and channel statistics every T time unit. Finally, we prove that the proposed scheduling achieves the exact $[O(1/V), O(V)]$

FIGURE 1: System model.

energy-queue tradeoff under wireless uplinks in the MCC system.

3. System Model and Problem Statement

In this section, we first present the model for resource-intensive application offloading in the MCC system. Then, we describe the formulation for the problem of energy-efficient for wireless uplinks. Finally, we employ the mHealth as an instance to show the process of the proposed algorithm.

3.1. System Model. We consider a MCC system with IaaS platform, which owns a server cluster to support mobile users. Without loss of generality, we use one server to represent the server cluster. When users are going to offload the mobile applications, the server will allocate appropriate amounts of virtual machines to execute these applications. For more details about this MCC system one can refer to [24–26]. Let T be the time unit; we use $T_k = \lceil t/T \rceil$ to denote the k th time unit, where t is any time. The MCC system operates in unit time. We consider a user who has N heterogeneous applications offloading to cloud through N time-varying channels. The data generated in each offloading application is processed in a corresponding queue, denoted by $\mathbf{Q}(T_k) = (Q_1(T_k), Q_2(T_k), \dots, Q_N(T_k))$, $T_k = 0, 1, \dots$, which is assumed to operate in a discrete time unit manner. For any application i , $Q_i(T_k)$ represents its queue backlog of data to be transmitted from the MD to the cloud at the beginning of the time unit T_k . Let $\mathbf{A}(T_k) = (A_1(T_k), A_2(T_k), \dots, A_N(T_k))$ be the vector denoting the amount of newly generated data of each offloading application in every time unit T_k , which is referred to as the vector data arrival rates that arrive in their corresponding queues $\mathbf{Q}(T_k)$. Suppose $A_i(T_k)$ is Poisson distributed with $\mathbb{E}[A_i(T_k)] = \lambda_i$; then we have $\mathbb{E}[\mathbf{A}(T_k)] = \boldsymbol{\lambda} \triangleq (\lambda_1, \lambda_2, \dots, \lambda_N)$. The system model is shown in Figure 1.

Let $\mathbf{r}(T_k) = (r_1(T_k), r_2(T_k), \dots, r_N(T_k))$ be the vector of current channel states for different channels during time unit T_k . After acquiring channel states, MD chooses appropriate queues to transmit their data packet in each time slot τ , where $\tau = T/k_1$, $k_1 \in \mathbb{Z}^+$. Let $\mu_{ij}(T_k) \in \{0, 1, \dots, \mu_{\max}\}$ be the service

rate of application i supported by channel j , where $\mu_{\max} \in \mathbb{Z}^+$. For the service rate of the queue i , we have

$$\mu_i(T_k) = \sum_{j=1}^N \mu_{ij}(T_k) w_{ij}(T_k), \quad (1)$$

where $w_{ij}(T_k)$ denotes the number of slots that the queue i is supported by channel j in time unit T_k .

According to [27], the power state of MD is divided into active state and idle state. Let $\alpha_i(T_k)$ be the power of the MD at the active state, which is supported by the channel j with current channel state $r_j(T_k)$ in time unit T_k . Let $\beta_i(T_k)$ be the idle power of the user in time unit T_k . Then the energy consumption of user in time unit T_k can be denoted as

$$\begin{aligned} E(T_k) \\ = \sum_{i=1}^N \sum_{j=1}^N [\alpha_i(T_k) w_{ij}(T_k) \tau + \beta_i(T_k) (T - w_{ij}(T_k) \tau)], \end{aligned} \quad (2)$$

where $\sum_{j=1}^N w_{ij}(T_k) \tau \leq T$.

3.2. Problem Statement. In the following, we assume that MD can estimate the unfinished delivery in its queues accurately. $\mathbf{Q}(T_k) = (Q_1(T_k), Q_2(T_k), \dots, Q_N(T_k))$ is the unfinished application data of the offloading in the MD at time unit T_k . We use the following queueing dynamics

$$Q_i(T_k + 1) = \max [Q_i(T_k) - \mu_i(T_k), 0] + A_i(T_k). \quad (3)$$

Throughout the paper, we require all the queues to be stable, which is defined as

$$\bar{Q} \triangleq \limsup_{T_k \rightarrow \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} \sum_{i=1}^N \mathbb{E} \{Q_i(s)\} < \infty, \quad (4)$$

where $\bar{Q}$ is the time average queue length and the expectation is taken over the randomness of $\mathbf{Q}(T_k)$.

In practice, dead spots or coverage holes usually cause disconnection (i.e., WLAN and 3G are not available). In this situation, a huge queue backlog even queue instability might be caused, because the mobile application cannot be offloaded to cloud any more. In fact, this problem can be tackled by making the cloud services temporarily stopped. We will treat this case as a joint optimization of local execution and transmitting cost in the future. Moreover, MCC faces the channel's competition problem, when several users offload the mobile applications to cloud at the same time. This problem has been treated in [28, 29] and is beyond the scope of this paper. In this paper, we only consider that a mobile user offloads mobile application from MD to cloud.

The focus of our work is energy optimal scheduling for data uploading with the time-varying wireless channel. We call every *feasible* policy that ensures (4) a *stable* policy and use E_{av}^* to denote the infimum average energy consumption

over all stable policies. Then, we define the time average energy consumption of a feasible policy Π as

$$E_{\text{av}}^{\Pi} \triangleq \limsup_{T_k \rightarrow \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} \mathbb{E} \{E^{\Pi}(s)\}, \quad (5)$$

where $E^{\Pi}(s)$ denotes the energy consumption on the MD by policy Π at time unit T_k .

The objective is to find a stable policy by reducing the number of transmission slots at every time unit T_k , so as to minimize the time average energy consumption of MD. We refer to this as the energy consumption minimization (ECM) problem in the remainder of the paper.

4. A Two-Time Dynamic Offloading Algorithm

In this section, we first analyze the ECM problem using Lyapunov optimization. Then, we describe T2DO algorithm and discuss corresponding insight and implementation-related issues.

4.1. The ECM Problem Analysis Using Lyapunov Optimization. We first define the Lyapunov function, $L(T_k)$, to measure the aggregate queue backlog in the system

$$L(T_k) \triangleq \sum_{i=1}^N \frac{1}{2} ([Q_i(T_k)]^2). \quad (6)$$

Next, we define the T -unit Lyapunov drift, $\Delta_T(T_k)$ as the expected change in the Lyapunov function over T_k units

$$\Delta_T(T_k) \triangleq \mathbb{E} \{L(T_k + 1) - L(T_k) \mid \mathbf{Q}(T_k)\}. \quad (7)$$

Following the Lyapunov optimization approach, we add the expected energy consumption with $V > 0$ over T_k units (i.e., a penalty function), $V\mathbb{E}\{E(T_k)\}$, to (7), which leads to the *drift-plus-penalty* term. This is a key step to obtain an upper bound on this term. Before further discussing the *drift-plus-penalty* term, we first present a lemma in [30], which is related to the derivation of the upper bound of *drift-plus-penalty* term.

Lemma 1. If $A, B, C, D \in \mathbb{R}^+$, and $A \leq \max[B - C, 0] + D$, then $A^2 \leq B^2 + C^2 + D^2 - 2B(C - D)$.

Based on (7) and Lemma 1, we derive a theorem, which characterizes the upper bound for our case.

Theorem 2. For any given $V > 0$, under any possible actions $\mu_{ij}(T_k) \in \{0, 1, \dots, \mu_{\max}\}$ and $A_i(T_k) \in \{0, 1, \dots, A_{\max}\}$, one has

$$\begin{aligned} \Delta_T(T_k) + V\mathbb{E} \{E(T_k) \mid \mathbf{Q}(T_k)\} \\ \leq B + V\mathbb{E} \{E(T_k) \mid \mathbf{Q}(T_k)\} \\ - \sum_{i=1}^N \mathbb{E} \{Q_i(T_k) [\mu_i(T_k) - A_i(T_k)] \mid \mathbf{Q}(T_k)\}, \end{aligned} \quad (8)$$

where $B = (N^3 \mu_{\max}^2 + N A_{\max}^2)/2$.

Input: $\mathbf{Q}(T_k)$
Output: $\mu_i(T_k)$

- (1) At the beginning of time unit T_k , monitor the queue backlog.
- (2) Monitor the channel states $\mathbf{r}(T_k)$ between cloud and MD.
- (3) The mobile agent chooses queue $Q_i(T_k)$ to transmit data to minimize

$$|Q_i(T_k) - Q_j(T_k)| \quad i \neq j, 0 \leq i, j \leq N \quad (*)$$
- (4) **if** $VE(T_k) - \sum_{i=1}^N Q_i(T_k)\mu_i(T_k) < 0$ **then**
- (5) The mobile agent chooses the channels to transmit data to minimize

$$VE(T_k) - \sum_{i=1}^N Q_i(T_k)\mu_i(T_k) \quad (**)$$
- (6) **else**
- (7) Stay idle for energy conservation.
- (8) **end if**
- (9) Update the queues using (1).

ALGORITHM 1: T2DO algorithm.

Proof. Squaring both sides of the queueing dynamic (1) and using Lemma 1, we have

$$[Q_i(T_k + 1)]^2 \leq [Q_i(T_k)]^2 + [\mu_i(T_k)]^2 + [A_i(T_k)]^2 - 2Q_i(T_k)[\mu_i(T_k) - A_i(T_k)]. \quad (9)$$

Summing (9) from $i = 1$ to N , using the fact that $\mu_i(T_k) \leq N\mu_{\max}$ and $A_i(T_k) \leq A_{\max}$, we obtain

$$\begin{aligned} & \sum_{i=1}^N ([Q_i(T_k + 1)]^2 - [Q_i(T_k)]^2) \\ & \leq N^3\mu_{\max}^2 + NA_{\max}^2 - 2\sum_{i=1}^N Q_i(T_k)[\mu_i(T_k) - A_i(T_k)]. \end{aligned} \quad (10)$$

Substituting (10) into (7), we get

$$\begin{aligned} \Delta_T(T_k) & \triangleq \mathbb{E}\{L(T_k + 1) - L(T_k) \mid \mathbf{Q}(T_k)\} \\ & \leq \sum_{i=1}^N \mathbb{E}\{Q_i(T_k)[A_i(T_k) - \mu_i(T_k)] \mid \mathbf{Q}(T_k)\} \\ & \quad + \frac{1}{2}(N^3\mu_{\max}^2 + NA_{\max}^2). \end{aligned} \quad (11)$$

Therefor by defining $B = (N^3\mu_{\max}^2 + NA_{\max}^2)/2$ and adding $V\mathbb{E}\{E(T_k) \mid \mathbf{Q}(T_k)\}$ to both sides of (11), we obtain (8). $\square$

4.2. The T2DO Algorithm Design (See Algorithm 1). According to [18], the design principle of Lyapunov framework is minimizing the upper bound of the *drift-plus-penalty* term; that is, in every time unit T_k , we try to choose a scheduling algorithm to minimize the RHS of (8). In fact, the scheduling algorithm only affects the energy consumption $E(T_k)$ and the queue i 's service rate $\mu_i(T_k)$ in the time unit T_k ; hence we

can minimize the RHS of (8) by minimizing the following simplified term

$$V\mathbb{E}\{E(T_k) \mid \mathbf{Q}(T_k)\} - \sum_{i=1}^N \mathbb{E}\{Q_i(T_k)\mu_i(T_k) \mid \mathbf{Q}(T_k)\}. \quad (12)$$

We now describe the system implementation of T2DO algorithm as follows.

The T2DO algorithm works at two different time scales. The MD assigns the service rate of application offloading at the beginning of every time unit T . Then MD chooses the channel to transmit data at every time τ slot. Two different time scales are important from an implementation perspective, because the MD's decision interval is usually much longer than channels' data transmission slot τ .

The proposed T2DO algorithm has two important properties. First, MD assigning decision remains unchanged for a time unit. We can properly increase T to reduce the computational overhead at the MD. Second, channel can defer the transmission of data if its channel state is too bad. Channel i may choose not to transmit data in a particular time slot τ or time unit T , even if $Q_i(T_k) > 0$, due to the low data transmission rates at the user.

4.3. A Practical Instance for T2DO Algorithm. In this section, we take the mHealth as an instance to show the process of the T2DO algorithm. The mHealth combines wearable body sensors [31] and MD (i.e., mobile-based medical monitoring device) to provide more effective and affordable healthcare services. Medical organizations focusing on hypertension, diabetes, geriatrics, or chronic diseases can take care of patients at their home instead of in the hospital. However, one of the disadvantages of eHealth is that different medical technologies are often supported by different vendors, which leads to poor interoperability. In addition, dedicated MDs are required. Constrained by computational power, storage space, and battery life of existing MDs, it is hard for them to execute resource-intensive applications. In this case, as an

FIGURE 2: MCC-based mHealth.

IaaS platform, cloud computing can contribute to offloading the eHealth services from MDs to the cloud. This paradigm not only improves the performance and the compatibility of MDs, but also raises the possibility of providing more accurate offsite personalized medical diagnosis and treatment [32].

In a MCC-based mHealth, as shown in Figure 2, the MD first connects to physiological body sensors to collect data, such as blood pressure, temperature, heart rate, and electrocardiograph. Then, the mobile device manages all sensing data acquired from wearable body sensors and uploads the physiological data to the cloud. Finally, the cloud stores and processes these physiological data. Once the diagnostic decision on the cloud is finished, the treatment plans will be sent back to the MD. In this case, our T2DO algorithm can reduce the energy consumed in data uploading between the MD and the cloud. This part of energy is the major energy consumption on MD for the mHealth.

Next, we show that how the T2DO algorithm reduces the energy consumption of uploading for the mHealth. Here, we also employ the mobile agent as an entity to manage and upload physiological data; thus the mobile agent is an executor of our T2DO algorithm. When the MD obtains physiological data from wearable body sensors, the mobile agent firstly estimates the queue backlog of these physiological data and monitors the channel state between the MD and the cloud at the beginning of time unit T_k . Secondly, the mobile agent chooses queue $Q_i(T_k)$ to transmit data to balance queue length among all queues. Thirdly, the mobile agent decides whether the data of queues be transmitted or not in current time unit and decides the amount of data of those queues which can be transmitted based on equality (**). Finally, according to the amount of transmitting and arriving data, the mobile agent updates the queues based on equality (3). According to above process of the T2DO algorithm, we can save the energy consumption on MD by

postponing data transmission when the channel states are terrible.

In this paper, we only take the mHealth as an example to show how our T2DO algorithm reduces the energy consumption of uploading. In fact, the T2DO algorithm can also be applied to other mobile applications such as mobile commerce, mobile learning, and mobile office.

5. Performance Analysis

According to [7], we characterize the optimal time average energy consumption E_{av}^* with below lemma, which can be achieved by any algorithm that stabilizes the queue.

Lemma 3. *For any data arrival rate vector $\lambda \in \Lambda$ and time unit T_k , there exists a stationary randomized control policy Π_{opt} that chooses appropriate queue to transmit its data packet in each slot τ ; then one has the following equalities*

$$\begin{aligned} \mathbb{E} \{E^{\Pi_{opt}}(T_k)\} &= E_{av}^*(\lambda) \\ \mathbb{E} \{\mathbf{A}(T_k)\} &= \mathbb{E} \left\{ \sum_{i=1}^N \mu_i^{\Pi_{opt}}(T_k) \right\}, \end{aligned} \quad (13)$$

where Λ denotes the capacity region of the system.

Lemma 3 shows that by using a stationary randomized algorithm, it is possible to achieve the minimum time average energy consumption E_{av}^* for a given data arrival rate vector λ .

Based on Theorem 2 and Lemma 3, we derive a theorem, which presents bounds on the time average energy consumption and queue backlogs achieved by T2DO algorithm.

Theorem 4. *Suppose there exists an $\epsilon > 0$ such that $\lambda + \epsilon \mathbf{1} \in \Lambda$; then under T2DO algorithm, the performance bounds of the*

time average energy consumption and queue backlog can be denoted as

$$\overline{Q^{T2DO}} \triangleq \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K \sum_{i=1}^N (\mathbb{E} \{Q_i(T_k)\}) \leq \frac{B + VE_{\max}}{\epsilon} \quad (14)$$

$$\overline{E^{T2DO}} \triangleq \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K \mathbb{E} \{E^{T2DO}(T_k)\} \leq E_{\text{av}}^*(\lambda) + \frac{B}{V}, \quad (15)$$

where $\mathbf{1}$ denotes the vector of all 1's and E_{av}^* and $E_{\max}$ are the optimal energy consumption and the maximum energy consumption for stationary randomized control, respectively.

Proof. Since $\lambda + \epsilon \mathbf{1} \in \Lambda$, it can be shown using Lemma 3 that there exists a stationary and randomized policy Π'_{opt} that achieves the following

$$\begin{aligned} \mathbb{E} \{E^{\Pi'_{\text{opt}}}(T_k)\} &= E_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) \\ \mathbb{E} \{A(T_k)\} &= \mathbb{E} \left\{ \sum_{i=1}^N \mu_i^{\Pi'_{\text{opt}}}(T_k) \right\} - \epsilon, \quad \forall i, \end{aligned} \quad (16)$$

where $E_{\text{av}}^*(\lambda + \epsilon \mathbf{1})$ is the optimal energy consumption corresponding to the rate vector $\lambda + \epsilon \mathbf{1} \in \Lambda$. Substituting (16) into (8), we get

$$\begin{aligned} \Delta_T(T_k) + V \mathbb{E} \{E^{T2DO}(T_k) \mid Q(T_k)\} \\ \leq B + VE_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) - \epsilon \left(\mathbb{E} \left\{ \sum_{i=1}^N Q_i(T_k) \mid Q(T_k) \right\} \right). \end{aligned} \quad (17)$$

Taking the expectation of (17) over $Q(T_k)$, we obtain

$$\begin{aligned} \mathbb{E} \{L(T_k + 1) - L(T_k)\} + V \mathbb{E} \{E^{T2DO}(T_k)\} \\ \leq B + VE_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) - \epsilon \left(\mathbb{E} \left\{ \sum_{i=1}^N Q_i(T_k) \right\} \right). \end{aligned} \quad (18)$$

Summing (18) from $T_k = 0$ to K , we have

$$\begin{aligned} \mathbb{E} \{L(T_k + 1)\} + V \mathbb{E} \{E^{T2DO}(T_k)\} \\ \leq KB + KVE_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) - \epsilon \sum_{T_k=1}^K \left(\mathbb{E} \left\{ \sum_{i=1}^N Q_i(T_k) \right\} \right). \end{aligned} \quad (19)$$

Then, using the fact that $E_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) < E_{\max}$, $\mathbb{E} \{L(T_k + 1)\} > 0$, and $V \mathbb{E} \{E^{T2DO}(T_k)\} > 0$ we have

$$\epsilon \sum_{T_k=1}^K \left(\mathbb{E} \left\{ \sum_{i=1}^N Q_i(T_k) \right\} \right) \leq KB + KVE_{\text{av}}^*(\lambda + \epsilon \mathbf{1}). \quad (20)$$

Dividing both sides of (20) by ϵK , we obtain

$$\frac{1}{K} \sum_{T_k=1}^K \left(\mathbb{E} \left\{ \sum_{i=1}^N Q_i(T_k) \right\} \right) \leq \frac{B + VE_{\max}}{\epsilon}. \quad (21)$$

Taking a lim sup as $K \rightarrow \infty$, (14) is proven.

Moreover, based on (19), using the fact that $\mathbb{E} \{L(T_k + 1)\} > 0$ and $\mathbb{E} \{\sum_{i=1}^N Q_i(T_k)\} > 0$, we can derive

$$V \mathbb{E} \{E^{T2DO}(T_k)\} \leq B + VE_{\text{av}}^*(\lambda + \epsilon \mathbf{1}). \quad (22)$$

Summing (22) from $T_k = 1$ to K and dividing both sides by KV , we have

$$\frac{1}{K} \mathbb{E} \left\{ \sum_{T_k=1}^K E^{T2DO}(T_k) \right\} \leq E_{\text{av}}^*(\lambda + \epsilon \mathbf{1}) + \frac{B}{V}. \quad (23)$$

Taking a lim sup as $K \rightarrow \infty$, using Lebesgue's dominated convergence theorem, and then letting $\epsilon \rightarrow 0$, we obtain inequality (15). $\square$

Equation (15) demonstrates that the energy consumption $\overline{E^{T2DO}}$ can be arbitrarily close to the optimum $E_{\text{av}}^*(\lambda)$ with an arbitrarily large V (i.e., approach $E_{\text{av}}^*(\lambda)$ within $O(1/V)$ deviation). At the same time, according to (14), the proposed algorithm guarantees an $O(V)$ average queue backlog. Thus, by appropriately selecting the control parameter V , we can achieve a desired tradeoff between energy consumption and queue backlog.

From the perspective of algorithm's performance, it is very important to value the parameter V . The following theorem gives the range of values for V .

Theorem 5. Assume that $\mu_{ij}(T_k) \in \{0, 1, \dots, \mu_{\max}\}$ is the service rate of application i supported by channel j , where $\mu_{\max} \in \mathbb{Z}^+$, E_T is the energy consumption of offload application during the time unit T_k , and the reasonable range of values allowed for V is $(0, M^2 \mu_{\max}^2 / E_T)$, where $M = T/\tau$.

Proof. Suppose there exists V' , MD transmits queue i 's data in the time unit T_k , and the service rate of the queue i is $\mu_i(T_k)$. According to the T2DO algorithm, we have

$$V' E_T - \sum_{i=1}^N Q_i(T_k) \mu_i(T_k) < 0. \quad (24)$$

Let some channel j be in the best state, its corresponding service rate $\mu_j(T_k) = (T/\tau) \mu_{\max}$, and we are setting $M = T/\tau$. If $\sum_{i=1}^N Q_i(T_k) < M \mu_{\max}$, then MD may defer a transmission for saving energy on the MD. Otherwise, MD chooses transmitting data. In this case, based on (24), we have

$$V' E_T - \sum_{i=1}^N Q_i(T_k) M \mu_{\max} \leq V' E_T - M \mu_{\max} M \mu_{\max} < 0. \quad (25)$$

Based on (25), we can derive $V' < M^2 \mu_{\max}^2 / E_T$. Moreover, according to Lyapunov optimization approach, we have $V' > 0$. Thus, we may safely draw the conclusion that the reasonable range of values allowed for V is $(0, M^2 \mu_{\max}^2 / E_T)$. $\square$

FIGURE 3: The impact of V on time average energy consumption and queue backlog.

6. Simulation Results

In this section, we take the mHealth as a data uploading scenario to evaluate the performance of the proposed algorithm, as shown in Figure 3. Let each MD upload physiological data to the cloud with four time-varying channels. According to [27], the powers of MD at active state and idle state are set to 1.680 W and 0.594 W, respectively. The average service rate of each channel is set to 1000 Kbit/s. The time unit T is set to 1 s; the slot τ is set to 0.1 s. We assume the data arrival rate of each application equals 800 Kbit/s. The simulation time K is set to 1000 s. Each Q_i is assumed to have an empty queue at the first time unit.

We first characterize the tradeoff between the energy consumption and the queue backlog for the T2DO algorithm. We plot the tradeoff between energy consumption and queue backlog for data transfer in Figure 3. It is clear that energy consumption falls quickly at the beginning and then tends to descend slowly while the time average queue backlog grows linearly with V . Hence, the variable V controls the energy-delay tradeoff of the T2DO algorithm. These results in Figure 3 are consistent with Theorem 4. Particularly, there exists a sweet spot of V (e.g., $V = 40000$), beyond which increasing V leads to little energy conservation yet significantly increases average queue backlogs.

Next, let V be fixed to 40000; we vary the data arrival rate from 0.3 Mbit/s to 0.6 Mbit/s. Figure 4 shows a comparison of energy consumption among our T2DO algorithm, random algorithm, SALSA algorithm, and minimum-delay algorithm. Comparing with the random algorithm and the minimum-delay algorithm, the energy consumption of the proposed T2DO algorithm has decreased by 30%. The reason is that our T2DO algorithm can find an expected channel with good condition to transmit the data by postponing the communication for a while. Furthermore, we also note that proposed T2DO algorithm can provide better performance than SALSA algorithm. This is because user can obtain data

FIGURE 4: Average energy consumption comparison among different algorithms.

FIGURE 5: Average queue backlog among different algorithms.

from any channel with better channel conditions in the proposed algorithm, while in SALSA algorithm user is only supported by a dedicated-channel.

Figure 5 shows a comparison of average queue backlog among four algorithms. We can see that the average queue backlog of proposed T2DO algorithm is longer than that of random algorithm and minimum-delay algorithm. It verifies that proposed T2DO algorithm can provide energy-delay tradeoff. Furthermore, we also note that proposed T2DO algorithm can provide better performance than SALSA algorithm in queue backlog. This is because the proposed T2DO algorithm can effectively increase service rate by dynamically

FIGURE 6: The $\bar{E}_e$ comparison among different algorithms.

choosing appropriate channels to transmit data for each time unit, and higher service rate results in lower queue backlog.

Finally, we compare the energy efficiency of T2DO algorithm with that of random algorithm, SALSA algorithm, and minimum-delay algorithm. For ease of comparison, we define a performance metric E_e , referred to as the average energy consumption per data packet

$$\bar{E}_e = \lim_{T \rightarrow \infty} \frac{\sum_{k=0}^T E(T_k)}{T \sum_{i=1}^N \lambda_i}. \quad (26)$$

Let the data arrival rate be fixed to 0.9 Mbit/s, and let the simulation time K be set to 1000 s. we vary V from 0 to 100000. Figure 6 compares the values of $\bar{E}_e$ under the four algorithms. It shows that the random algorithm and minimum-delay algorithm consume about 1.5 Joules to transmit per Mbit data. We can see that the $\bar{E}_e$ of T2DO algorithm and SALSA algorithm is lower than that of random algorithm and minimum-delay algorithm. Since T2DO algorithm and SALSA algorithm are designed according to Lyapunov optimization, they can transmit the data with good condition by postponing the communication for a while. It is interesting to note that the $\bar{E}_e$ of T2DO algorithm and SALSA algorithm tends to decrease as the parameter V increases, and T2DO algorithm can provide better performance than SALSA algorithm. The reason is that the parameter V control the tradeoff between energy consumption and queue backlog, so the energy consumption will decrease when V increases. Moreover, T2DO algorithm can effectively reduce the queue backlog by transmitting data over any channel with better channel conditions; thus for the same parameter V , our T2DO algorithm has the lower energy consumption per data than SALSA algorithm.

7. Conclusion

In this paper, we investigate the energy-efficient problem about the uploading data when applications are offloaded with a time-varying channel scenario in MCC. By the Lyapunov optimization, a T2DO algorithm is presented, where the tradeoff between energy and queue backlog for offloading is achieved. This T2DO algorithm first allocates data to the appropriate queuing and then chooses the appropriate channel to transmit their data packet with the queue backlog and channel statistics every T time unit. Moreover, we take the mHealth as an example and show the process of the proposed algorithm in real applications. The simulation results fully demonstrate the correctness and effectiveness of the proposed algorithm.

Unfortunately, this study also suffers from some practical issues. For instance, the delay constraint is another key factor of concern for mobile application, besides computational performance and energy efficiency. Moreover, different mobile applications have delay requirements. In the follow-up work, we will characterize the energy consumption caused by various applications uploading and explore alternative ways to reduce transmission energy, while meeting their delay requirements. We also plan to evaluate the performance of our T2DO algorithm by running real mobile applications, such as mHealth, mobile office, mobile commerce, and mobile learning.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Nature Science Foundation of China under Grant no. 61173017 and the National High-Tech R&D Program under Grant no. 2014AA01A701.

References

- [1] F. Liu, P. Shu, H. Jin et al., "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–21, 2013.
- [2] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1334–1354, 2013.
- [3] H. B. Liang, L. X. Cai, D. J. Huang, X. M. Shen, and D. Y. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 2222–2232, 2012.
- [4] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.
- [5] M. Zafer and E. Modiano, "Minimum energy transmission over a wireless channel with deadline and power constraints," *IEEE*

- Transactions on Automatic Control*, vol. 54, no. 12, pp. 2841–2852, 2009.
- [6] F. W. Fu and M. van der Schaar, "Structure-aware stochastic control for transmission scheduling," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 3931–3945, 2012.
 - [7] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.
 - [8] C.-P. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 527–539, 2010.
 - [9] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Communications Letters*, vol. 3, no. 2, pp. 153–156, 2014.
 - [10] M. Altamimi, A. Abdrabou, S. Naik, and A. A. Nayak, "Energy cost models of smartphones for task offloading to the cloud," *IEEE Transactions on Emerging Topics Computing*, vol. 6, no. 1, pp. 1–14, 2015.
 - [11] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 255–269, San Francisco, Calif, USA, June 2010.
 - [12] U. Mandal, M. Habib, S. Zhang, B. Mukherjee, and M. Tornatore, "Greening the cloud using renewable-energy-aware service migration," *IEEE Network*, vol. 27, no. 6, pp. 36–43, 2013.
 - [13] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 5, pp. 1045–1057, 2014.
 - [14] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, "Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2648–2660, 2014.
 - [15] W. Zhang, Y. Wen, and D.-O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 81–93, 2015.
 - [16] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.
 - [17] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.
 - [18] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
 - [19] E. Tilevich and Y. W. Kwon, "Cloud-based execution to improve mobile application energy efficiency," *Computer*, vol. 47, no. 1, pp. 75–77, 2014.
 - [20] E. Katranaras, M. A. Imran, M. Dianati, and R. Tafazolli, "Green inter-cluster interference management in uplink of multi-cell processing systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 6580–6592, 2014.
 - [21] G. Miao, N. Himayat, G. Y. Li, and S. Talwar, "Low-complexity energy-efficient scheduling for uplink OFDMA," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 112–120, 2012.
 - [22] F. Liu, K. Zheng, W. Xiang, and H. Zhao, "Design and performance analysis of An energy-efficient uplink carrier aggregation scheme," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 2, pp. 197–207, 2014.
 - [23] S. Deb and P. Monogioudis, "Learning-based uplink interference management in 4G LTE cellular systems," *IEEE/ACM Transactions on Networking*, vol. 11, no. 9, pp. 1063–1081, 2014.
 - [24] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.
 - [25] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 337–368, 2014.
 - [26] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: on networking and computing environments' integration," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, 2013.
 - [27] S. Berger, B. Almeroth, V. Suryaprakash, P. Zanier, I. Viering, and G. Fettweis, "Dynamic range-aware uplink transmit power control in LTE networks: establishing an operational range for LTE's open-loop transmit power control parameters (α, P_0)," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 521–524, 2014.
 - [28] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.
 - [29] B. Cao, G. Feng, Y. Li, and C. Wang, "Cooperative media access control with optimal relay selection in error-prone wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 252–265, 2014.
 - [30] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFO-backpressure achieves near-optimal utility-delay trade-off," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, 2013.
 - [31] B. Cao, Y. Ge, C. W. Kim, G. Feng, H. P. Tan, and Y. Li, "An experimental study for inter-user interference mitigation in wireless body sensor networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3585–3595, 2013.
 - [32] X. L. Wang, Q. Gui, B. W. Liu, Z. P. Jin, and Y. Chen, "Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ECG telemonitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 739–745, 2014.