

Combined Service Subscription and Delivery Energy-Efficient Scheduling in Mobile Cloud Computing

Xing Liu¹, Chaowei Yuan¹, Enda Peng¹, and Zhen Yang²

¹School of Information and Communication Engineering, Beijing University
of Post and Telecommunications, Beijing, 100876, P. R. China
[e-mail: {liuxing99, yuancw200}@bupt.edu.cn; pengenda@gmail.com]

²School of Computer Science, Beijing University of Posts and Telecommunications
Beijing, 100876, P. R. China [e-mail: yangzhen@bupt.edu.cn]

*Corresponding author: Xing Liu

*Received July 29, 2014; revised January 27, 2015; revised March 11, 2015; accepted April 19, 2015;
published May 31, 2015*

Abstract

Mobile cloud computing (MCC) combines mobile Internet and cloud computing to improve the performance of applications. In MCC, the data processing and storage for mobile devices (MDs) is provided on the remote cloud. However, MCC faces the problem of energy efficiency caused by randomly varying channels. In this paper, by introducing the Lyapunov optimization method, we propose a combined service subscription and delivery (CSSD) algorithm that can guide the users to subscribe to services reasonably. This algorithm can also determine whether to deliver the data and to whom data is sent in the current time unit based on the queue backlog and the channel state. Numerical results validate the correctness and effectiveness of our proposed CSSD algorithm.

Keywords: Mobile cloud computing, mobile agent, queueing, energy efficiency, Lyapunov optimization

1. Introduction

Mobile cloud computing (MCC) is regarded as an emerging research area that could combine cloud computing [1] with the mobile Internet [2]. MCC can improve the performance of mobile applications that are offered by the mobile cloud service providers by offloading data processing from mobile devices (MDs) to the remote cloud. As a result, MDs do not need a powerful configuration since all the complicated computing can be processed in the cloud servers [3]. However, the amount of application data that is transmitted on both wired and wireless networks increases rapidly because all users' requests have to be supported from the remote cloud [4, 5]. As a result, the communication overhead of MD will consume a significant amount of battery energy. Moreover, the stochastic wireless channels and the MDs leave the network might lead to link loss, which can cause serious energy consumption [6]. Therefore, reducing the energy consumption of transmission is one of the most significant issues in MCC.

The energy efficiency of data transmission in wireless networks was investigated in [7, 8], [9, 10]. In [7], a code-based sleep and wake-up scheduling scheme was presented to minimize energy consumption caused by idle listening. In [8], an optimal sleep interval was derived to minimize power consumption while still satisfying the mouth-to-ear delay constraint. In [9], an enhanced dynamic power management method was proposed to schedule scaled jobs at slack time with the goal of saving energy and keeping system reliability. These studies focused on controlling sleep interval during the sleep state, while Choi and Lee [10] investigated how to efficiently control the transition from active to sleep state by considering the attributes of the request-and-response delay. However, the above proposals only considered the power states of the MDs during data transmission, ignoring the time-varying channel environment. Poor channel conditions result in an increase in data transmission time [11], and this causes an increase in transmission energy.

In order to further reduce transmission energy in the time-varying channel environment, some transmission schemes [12, 13, 14] have been studied. Zafer and Modiano [12] used a novel continuous-time optimal-control formulation and Lagrangian duality to propose an optimal transmission scheduling that can dynamically adapt to time-varying channels to minimize energy consumption. Neely [13] exploited time-varying channel conditions to design the Energy-Efficient Control (EEC) algorithm based on the Lyapunov optimization method. Ra et al. [14] implemented the Lyapunov optimization method on MDs in multiple wireless interface environments and proposed the Stable and Adaptive Link Selection Algorithm (SALSA) with time-varying V values to meet different video application delay tolerances. In fact, the methods of [12, 13, 14] find an expected channel with good condition to transmit the data by postponing the communication for a period of time such that a balance between the energy consumption and queue backlog for mobile applications is achieved. In practice, the channel conditions of MDs may become extremely poor over a very long period of time because of coverage holes or pilot pollution. In this situation, on the one hand, channels may defer the data transmission until their supported user's channel condition is sufficiently good. On the other hand, as the queue backlog increases, channels may transmit the data to their support user at the low service rate. Indeed, a balance between energy consumption and queue backlog can be achieved. However, to achieve this, a huge energy consumption is incurred.

Recently, some researchers have proposed the idea of proactive fetching applications for all subscribed mobile users [15, 16]. Motivated by them, in this paper, we address the optimal transmission energy for the time-varying channel scenario in MCC networks that takes into account the number of service subscriptions. By introducing the Lyapunov optimization method, we propose a combined service subscription and delivery (CSSD) algorithm that can guide a user to subscribe to services reasonably and determine whether to deliver the data as well as to whom this data is sent in the current time unit based on the queue backlog and channel state. Numerical results validate that our proposed CSSD algorithm can effectively decrease both the transmission energy and queue backlog by making transmission decisions. As a result, users are guided to rational cloud service subscription choices.

The rest of this paper is organized as follows. The system model and the problem statement are introduced in Section 2. In Section 3, the proposed CSSD algorithm is explained in detail. In Section 4, the performance of the CSSD algorithm is analyzed. Section 5 presents experimental results and performance comparisons. Finally, Section 6 concludes the paper.

2. System Model and Problem Statement

We consider an MCC environment in which a mobile application is divided into two parts [17], i.e., communications and cloud computing (e.g., CPU, memory, storage, and applications). The mobile agent, which resides at the base station, is expected to bridge the gap between cloud computing and data transmission that traditionally operated quite separately [18]. In this paper, we suppose that cloud services can be promoted to users through a mobile agent and service data can be delivered to users via a time-varying channel. The system model is shown in Fig. 1.

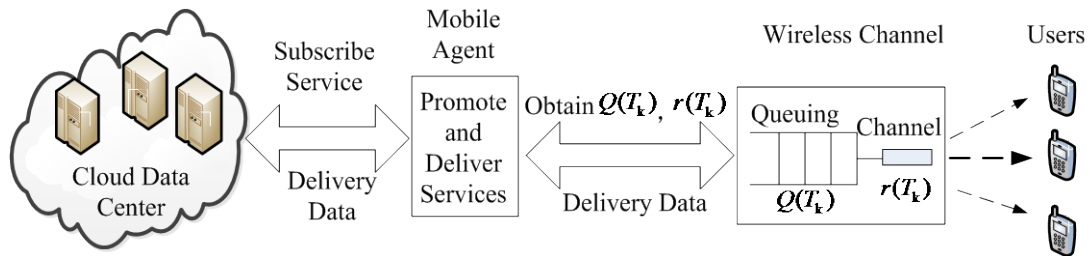


Fig. 1. System model

2.1 System Model

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 system operates in unit time. Define $a(T_k)$ to be the amount of data arriving at a mobile agent when the system runs a mobile application (e.g., video streaming, audio streaming, or news reading) during time unit T_k . Suppose $a(T_k)$ is Poisson distributed with $E[a(T_k)] = \lambda$ and $a(T_k) \in \{0, 1, \dots, A_{\max}\}$. Let user $i \in \{0, 1, \dots, N\}$ subscribe to n_i mobile applications in time unit T_k , then the number of data packets arriving at the mobile agent is $n_i a(T_k)$ during this time unit, where $0 \leq n_i \leq n_{\max}$. The data packet of user i that arrives at the mobile agent is queued in Q_i .

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 users during time unit T_k . After acquiring these channel states, the channel chooses the appropriate user to transmit their data packet to in each time slot τ , where $\tau = T/k_1$, $k_1 \in \mathbb{Z}^+$. An example of the different time scales τ and T is given in Fig. 2. In this example, $k_1=10$, thus $T=10\tau$. Let $\mu^i(T_k) \in \{1, 2, \dots, \mu_{\max}\}$ be the transmission rate of user i supported by the channel, where $\mu_{\max} \in \mathbb{Z}^+$. Therefore, the service rate of user i supported by the channel can be denoted as $\mu_i(T_k) = w_i(T_k)\mu^i(T_k)\tau$, where $w_i(T_k)$ denotes the number of slots that the channel supports for user i in time unit T_k . We then have the service rate of the channel: $\mu(T_k) = \sum_{i=1}^N \mu_i(T_k)$.

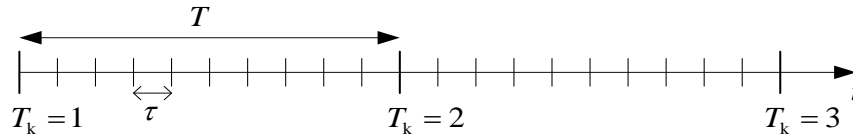


Fig. 2. Example of different time scales τ and T .

According to [19], the power states of an MD are divided into the active and idle states. Let $\alpha_i^{r_i(T_k)}$ be the power of user i in the active state, which is supported by the channel with current channel state $r_i(T_k)$ in time unit T_k . Let $\beta_i(T_k)$ be the idle power of user i in time unit T_k . We can then use $E(T_k) = \sum_{i=1}^N [\alpha_i^{r_i(T_k)} w_i(T_k)\tau + \beta_i(T_k)(T - w_i(T_k)\tau)]$ to denote the energy consumption of all users at time unit T_k , where $T \geq \sum_{i=1}^N w_i(T_k)\tau$.

2.2 Problem Statement

In the following, we assume that a mobile agent can estimate the unfinished delivery in a channel’s queues accurately. Let $\mathbf{Q}(T_k) = (Q_1(T_k), Q_2(T_k), \dots, Q_N(T_k))$, $T_k = 1, 2, \dots$ be the vector denoting user data queued in \mathbf{Q} at time unit T_k , We use the following queueing dynamics:

$$Q_i(T_k + 1) = \max[Q_i(T_k) - \mu_i(T_k), 0] + n_i a(T_k) \tag{1}$$

Throughout the paper, we use the following definition of queue stability:

$$\overline{Q} = \limsup_{T_k \rightarrow \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} \sum_{i=1}^N E[Q_i(s)] < \infty \tag{2}$$

The focus of our work is energy optimal scheduling for time-varying wireless channels. We call every *feasible* policy that ensures (2) a *stable* policy, and use E_{av}^* to denote the infimum average energy consumption over all stable policies. We then define the time-averaged energy consumption of a feasible policy Π as:

$$E_{\text{com}}^\Pi = \limsup_{T_k \rightarrow \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} E[E^\Pi(s)] \tag{3}$$

where $E^\Pi(s)$ denotes the energy consumption of all users by policy Π at time unit T_k .

The objective of our problem is to find a stable policy by which the mobile agent promotes cloud services to users properly, and the policy makes transfer decision every T time units in terms of different channel states and queue backlog to minimize the time-averaged energy consumption.

3. CSSD Algorithm

In this section, we consider cloud services that can be promoted to users by a mobile agent and present the CSSD algorithm based on the Lyapunov optimization method.

Based on the Lyapunov optimization method, our CSSD algorithm can reduce energy consumption while maintaining queue stability. We first define the Lyapunov function $L(T_k)$ to measure the aggregate queue backlog in the system as:

$$L(T_k) = \frac{1}{2} \sum_{i=1}^N [Q_i(T_k)]^2 \quad (4)$$

We next define the T -unit Lyapunov drift $\Delta_T(T_k)$ as the expected change in the Lyapunov function over T_k units.

$$\Delta_T(T_k) = E\{L(T_k + 1) - L(T_k) | \mathbf{Q}(T_k)\} \quad (5)$$

Following the Lyapunov optimization method, we add the expected energy consumption $VE\{E(T_k)\}$, where $V > 0$ over T_k units (*i.e.*, a penalty function), to (5), which leads to the *drift-plus-penalty* term. This is a key step to obtaining an upper bound on this term. The following lemma gives the upper bound for our case.

Lemma 1: For any given $V > 0$, under any possible actions $\mu_i(T_k) \in \{1, 2, \dots, \mu_{\max}\}$ and $a(T_k) \in \{0, 1, \dots, A_{\max}\}$ we have:

$$\Delta_T(T_k) + VE\{E(T_k)\} \leq B + VE\{E(T_k)\} - \sum_{i=1}^N E\{Q_i(T_k)\mu_i(T_k) | \mathbf{Q}(T_k)\} + \sum_{i=1}^N E\{Q_i(T_k)n_i a(T_k) | \mathbf{Q}(T_k)\} \quad (6)$$

where $B = \frac{N\mu_{\max}^2 + N\lambda^2 n_{\max}^2}{2}$.

Proof: Squaring both sides of the queueing dynamic (1) and using the fact that for any $x \in \mathbb{R}$, $((\max[x, 0])^2 \leq x^2)$, we have:

$$[Q_i(T_k + 1)]^2 \leq [Q_i(T_k)]^2 + [\mu_i(T_k)]^2 + [n_i a(T_k)]^2 - 2Q_i(T_k)[\mu_i(T_k) - n_i a(T_k)] \quad (7)$$

Inserting (4) and (7) into (5), we get:

$$\Delta_T(T_k) \leq \frac{1}{2} \sum_{i=1}^N E\{[\mu_i(T_k)]^2 + [n_i a(T_k)]^2 - 2Q_i(T_k)[\mu_i(T_k) - n_i a(T_k)]\} \quad (8)$$

Using the fact that $\mu_i(T_k) \leq \mu_{\max}$ and $0 \leq n_i \leq n_{\max}$, we have:

$$\Delta_T(T_k) \leq \frac{N\mu_{\max}^2 + N\lambda^2 n_{\max}^2}{2} - \sum_{i=1}^N E\{Q_i(T_k)\mu_i(T_k) | \mathbf{Q}(T_k)\} + \sum_{i=1}^N E\{Q_i(T_k)n_i a(T_k) | \mathbf{Q}(T_k)\} \quad (9)$$

Therefore, by defining $B = \frac{N\mu_{\max}^2 + N\lambda^2 n_{\max}^2}{2}$, and adding $VE\{E(T_k)\}$ to both sides of (9), we obtain equation (6).

According to [20], the design principle behind the Lyapunov optimization method is to minimize the upper bound of the *drift-plus-penalty* term, i.e., in every time unit T_k , we attempt to choose a control policy to minimize the right hand side of (6). The channel's transfer decision can only affect the energy consumption $E(T_k)$ and the service rate of each user

$\mu_i(T_k)$ during time unit T_k , hence, we can minimize $VE\{E(T_k)\} - \sum_{i=1}^N E\{n_i a(T_k) | \mathbf{Q}(T_k)\}$ by

the channel's transfer decision. Note that cloud service providers want users to subscribe to services as much as possible. As the user subscriptions increase, the service data also increases. This causes a large queue backlog and high energy consumption. Fortunately, in a practical application, the mobile agent can promote cloud services to users based on the queue backlog and channel state. Therefore, users can subscribe to cloud services reasonably based on the mobile agent's promotion information. We assume here that the control policy can minimize the right hand side of (6) when

$VE\{E(T_k)\} - \sum_{i=1}^N E\{Q_i(T_k)\mu_i(T_k) | \mathbf{Q}(T_k)\} + \sum_{i=1}^N E\{n_i a(T_k) | \mathbf{Q}(T_k)\} = 0$. The optimal

subscription solution is then $n_{\text{opt}} = \frac{\sum_{i=1}^N Q_i(T_k)\mu_i(T_k) - VE(T_k)}{\lambda \sum_{i=1}^N Q_i(T_k)}$, $n_{\text{opt}} \geq 0$.

Based on the above analysis, our CSSD algorithm is composed of transfer decisions, calculation of the optimal subscription solution, collection of the subscribed services, and queues updating. Clearly, the mobile agent is the executor of the proposed CSSD algorithm. We now describe the mobile agent implementation of the CSSD algorithm as follows.

Step 1: At every T_k , the channel chooses the user's data to transmit in order to minimize:

$$VE\{E(T_k)\} - \sum_{i=1}^N E\{n_i a(T_k) | \mathbf{Q}(T_k)\} \quad (10)$$

Step 2: At every T_k , the mobile agent promotes cloud services to users with

$$\max \left[\frac{\sum_{i=1}^N Q_i(T_k)\mu_i(T_k) - VE(T_k)}{\lambda \sum_{i=1}^N Q_i(T_k)}, 0 \right].$$

Step 3: The mobile agent collects subscribed services from the users.

Step 4: The mobile agent requests the cloud data center to initiate the subscribed services, and update the queues using (1).

In Step 1, the optimization objective of (10) is to minimize the number of transmission slots. This requires the mobile agent to detect all channel queue lengths every T time units, which only requires a few calculations and takes a very short time. Moreover, it requires the mobile agent to know the channel state information for each user. In the proposed method, we assume that the mobile agent can acquire each user's transmission rate based on the channel state information. As we only execute the channel's transfer decision that minimizes the expression

$VE\{E(T_k)\} - \sum_{i=1}^N E\{n_i a(T_k) | \mathbf{Q}(T_k)\}$, the transmission takes place only if some transfer decision $VE\{E(T_k)\} - \sum_{i=1}^N E\{n_i a(T_k) | \mathbf{Q}(T_k)\} < 0$. This happens when either user i 's channel condition is good enough, achieving a large service rate $\mu_i(T_k)$, or the queue $Q_i(T_k)$ is already congested at time unit T_k . At the beginning of time unit T_k , the mobile agent has not only determined whether to deliver the data or not, but also decided when to send and whom to send the data to. If these are some users not assigned to transmission slots in the current transfer decision, the mobile agent will notify these users to go to or maintain a sleeping status to save energy. In Step 2, the mobile agent promotes cloud services to users with the optimal subscription amount at time unit T_k . In a practical application, users subscribing to services simply refer to this optimal subscription solution. Therefore, there is some deviation between subscribed services and the optimal subscription solution in time unit T_k . The possible range and impact of this deviation is discussed in next section. In Step 3, as the mobile agent collects the subscribed service information from users, the mobile agent becomes a means by which users are notified of the new content of services. This new content is composed of the titles and number of services recommended for subscription. The users utilize this new content as a reference when subscribing to appropriate services. In Step 4, according to the subscribed service information from users, the mobile agent first notifies the cloud data center to initialize subscribed services. The mobile agent then collects and stores users' data packets in the relevant queues.

4. Performance Analysis

In this section, we first analyze the performance guarantee provided by our CSSD algorithm. We then explain the possible range and impact of the deviation between subscribed services and the optimal subscription solution.

4.1 Bound of Queue Backlog and Energy Consumption

According to [21], we characterize the optimal time-averaged energy consumption $E_{av}^*(T_k)$ with the following lemma that can be achieved by any algorithm that stabilizes the queue.

Lemma 2: For any rate vector $\lambda \subset \Lambda$ and time unit T_k , there exists a stationary randomized control policy Π_{opt} that chooses the appropriate user to transmit its data packet in each τ slot. We then have the following equalities:

$$E\{E^{\Pi_{opt}}(T_k)\} = E_{av}^*(\lambda) \quad (11)$$

$$\sum_{i=1}^N E\{\lambda_i\} = \sum_{i=1}^N E\{\mu_i^{\Pi_{opt}}(T_k)\} \quad (12)$$

where Λ denotes the capacity region of the system.

Lemma 2 shows that using a stationary randomized algorithm, it is possible to achieve the minimum time-averaged energy consumption $E_{av}^*(\lambda)$ for a given data packet arrival rate vector $\lambda = \{\lambda_1, \lambda_2 \dots \lambda_N\}$ where $\lambda_i = n_i E\{a(T_k)\}$.

Based on Lemmas 1 and 2, we derive a theorem that presents the bounds on the time-averaged energy consumption and queue backlog achieved by the CSSD algorithm.

Theorem 1: Suppose there exists an $\varepsilon > 0$ such that $\lambda^* + \varepsilon \mathbf{1} \in \Lambda$, then the performance bounds of the time-averaged energy consumption and queue backlog can be calculated as:

$$\overline{Q^{\text{CSSD}}} = \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K \sum_{i=1}^N \mathbb{E}\{Q_i(T_k)\} \leq \frac{B + VE_{\max}}{\varepsilon} \quad (13)$$

$$\overline{E^{\text{CSSD}}} = \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K \mathbb{E}\{E(T_k)\} \leq \frac{B}{V} + \mathbb{E}\{E_{\text{av}}^*(\lambda^*)\} \quad (14)$$

Here, $\mathbf{1}$ denotes the vector of all 1's, $\lambda^* = \{\lambda_1^*, \lambda_2^* \cdots \lambda_N^*\}$ such that $\sum_{i=1}^N \lambda_i^* = n_{\text{opt}} \mathbb{E}\{a(T_k)\}$,

and $E_{\text{av}}^*(\lambda^*)$ and E_{\max} are the optimal and maximum energy consumptions for stationary randomized control policy, respectively.

Proof: Because $\lambda^* + \varepsilon \mathbf{1} \in \Lambda$, it can be shown using Lemma 2 that there exists a stationary and randomized policy Π_{opt} that achieves the following:

$$\mathbb{E}\{E^{\Pi_{\text{opt}}}(T_k)\} = E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1}) \quad (15)$$

$$\sum_{i=1}^N \mathbb{E}\{n_i^{\Pi_{\text{opt}}} a(T_k)\} = \sum_{i=1}^N \mathbb{E}\{\mu_i^{\Pi_{\text{opt}}}(T_k)\} - \varepsilon \quad (16)$$

where $E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1})$ is the optimal energy consumption corresponding to the rate vector $\lambda^* + \varepsilon \mathbf{1} \in \Lambda$. Substituting (15) and (16) into (6), we get:

$$\Delta_T(T_k) + V\mathbb{E}\{E^{\text{CSSD}}(T_k) | \mathbf{Q}(T_k)\} \leq B + V\mathbb{E}\{E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1}) | \mathbf{Q}(T_k)\} - \varepsilon \mathbb{E}\{\sum_{i=1}^N Q_i(T_k) | \mathbf{Q}(T_k)\} \quad (17)$$

Substituting (5) into (17) and taking the expectation over $\mathbf{Q}(T_k)$, we obtain:

$$\mathbb{E}\{L(T_k + 1) - L(T_k)\} + \varepsilon \mathbb{E}\{\sum_{i=1}^N Q_i(T_k)\} \leq B + V\mathbb{E}\{E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1})\} \quad (18)$$

Summing (18) from $T_k = 0$ to $K - 1$, using the fact that $E^*(\lambda^* + \varepsilon \mathbf{1}) \leq E_{\max}$, we have:

$$\varepsilon \mathbb{E}\{\sum_{T_k=0}^K \sum_{i=1}^N Q_i(T_k)\} \leq KB + VK\mathbb{E}\{E_{\max}\} + \mathbb{E}\{L(0)\} \quad (19)$$

Dividing both sides of (19) by εK , we obtain:

$$\frac{1}{K} \sum_{T_k=1}^K \mathbb{E}\{\sum_{i=1}^N Q_i(T_k)\} \leq \frac{B + VE_{\max}}{\varepsilon} \quad (20)$$

Taking the limit superior as $K \rightarrow \infty$, (13) is proven.

Moreover, based on (17), we can derive:

$$V\mathbb{E}\{E^{\text{CSSD}}(T_k) | \mathbf{Q}(T_k)\} \leq B + V\mathbb{E}\{E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1}) | \mathbf{Q}(T_k)\} \quad (21)$$

Taking expectation of (21) over $\mathbf{Q}(T_k)$, then summing $T_k = 1$ to K , we have:

$$V\mathbb{E}\{\sum_{T_k=1}^K E^{\text{CSSD}}(T_k)\} \leq KB + KV\mathbb{E}\{E_{\text{av}}^*(\lambda^* + \varepsilon \mathbf{1})\} \quad (22)$$

Dividing both sides of (22) by KV , we obtain:

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

Taking the limit superior as $K \rightarrow \infty$, using the Lebesgue's dominated convergence theorem, and then letting $\varepsilon \rightarrow \infty$, we obtain inequality (14).

The proposed CSSD algorithm has two important properties. First, the mobile agent guides users to reasonably subscribe to services based on the queue backlog and channel state. It could also avoid a situation where channels transmit data to users with a low service rate when the queue backlog is sufficiently high. Therefore, our CSSD algorithm not only can reduce the time-averaged energy consumption, but also can decrease the time-averaged queue backlog. Second, our CSSD algorithm is designed based on the Lyapunov optimization method that reduces energy consumption by guiding users to subscribe to services reasonably. Nevertheless, the EEC algorithm [13] may defer the data transfer until the channel's transfer rate or queue backlog is sufficiently high. The method aims to achieve a balance between energy consumption and queue backlog by designing V values. The latter is a scheduling algorithm, and the former combines service subscriptions and dynamic scheduling based on the latter at every time unit.

4.2 Deviation of subscribed services

In a practical application, there is a deviation between the subscribed services and optimal subscription solution for our CSSD algorithm. The following theorem gives the possible range of this deviation.

Theorem 2: Assume that $\Delta n_{\text{total}}(T_k)$ is the deviation between the subscribed services and the optimal subscription solution, and let μ_{max} be the maximum service rate of the channel at time

unit T_k . In the CSSD algorithm, the possible range of $\Delta n_{\text{total}}(T_k)$ is then $[-n_{\text{opt}}, \frac{\mu_{\text{max}}}{\lambda} - n_{\text{opt}}]$,

$$\text{where } n_{\text{opt}} = \max \left[\frac{\sum_{i=1}^N Q_i(T_k) \mu_i(T_k) - VE(T_k)}{\lambda \sum_{i=1}^N Q_i(T_k)}, 0 \right].$$

Proof: For any time unit T_k , let n_{real} and n_{opt} be the subscribed services and optimal subscription solution, respectively. The deviation between the subscribed services and optimal subscription solution can then be described as:

$$\Delta n_{\text{total}}(T_k) = n_{\text{real}}(T_k) - n_{\text{opt}} \quad (24)$$

Moreover, since μ_{max} is the maximum service rate of the channel and $E\{a(T_k)\} = \lambda$, then the

maximum subscribed services is $\frac{\mu_{\text{max}}}{\lambda}$. Combining this value with the subscribed services should not be less than zero, hence we get:

$$0 \leq n_{\text{real}} \leq \frac{\mu_{\text{max}}}{\lambda} \quad (25)$$

Inserting (25) into (24), we find that the possible range of $\Delta n_{\text{total}}(T_k)$ is $[-n_{\text{opt}}, \frac{\mu_{\text{max}}}{\lambda} - n_{\text{opt}}]$.

According to the CSSD algorithm, when $\Delta n_{\text{total}}(T_k) \leq 0$, i.e., the subscribed services n_{real} are less than the optimal subscription solution n_{opt} , the channel can postpone communication for a period of time to wait for a better expected channel condition because the queue backlog is not big enough. Therefore, in this case, our CSSD algorithm has low energy consumption and queue backlog. When $\Delta n_{\text{total}}(T_k) > 0$, the queue backlog can become so large during time unit T_k , the channel can deliver the data packets whatever the channel condition is to service users with low transmission delay. In this case, high battery consumption caused by the low transmission rate is incurred. According to the above analyses, the energy consumption and queue backlog would be affected if $\Delta n_{\text{total}}(T_k) > 0$. The following theorem shows that the impact of the CSSD algorithm on the time-averaged energy consumption and queue backlog by $\Delta n_{\text{total}}(T_k) > 0$.

Theorem 3: Let $\Delta n_{\text{total}}(T_k) = \sum_{i=1}^N \Delta n_i(T_k) > 0$, $E\{\Delta n_i(T_k)\} = d_i$, suppose there exists an $\varepsilon > 0$ such that $\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1} \subset \boldsymbol{\Lambda}$, then the performance bounds of the time-averaged energy consumption and queue backlog can be calculated as:

$$\overline{Q^{\text{CSSD}}} = \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K \sum_{i=1}^N E\{Q_i(T_k)\} \leq \frac{B + d_{\text{total}} Q_{\text{max}} + V E_{\text{max}}}{\varepsilon} \quad (26)$$

$$\overline{E^{\text{CSSD}}} = \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{T_k=1}^K E\{E(T_k)\} \leq \frac{B + d_{\text{total}} Q_{\text{max}}}{V} + E\{E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d})\} \quad (27)$$

where $\boldsymbol{\lambda}^* = \{\lambda_1^*, \lambda_2^* \cdots \lambda_N^*\}$ such that $\sum_{i=1}^N \lambda_i^* = n_{\text{opt}} E\{a(T_k)\}$, $\mathbf{d} = (d_1, d_2 \cdots d_n)$,

$d_{\text{total}} = \sum_{i=1}^N d_i$, and Q_{max} , E_{max} , and $E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d})$ are the maximum queue backlog,

maximum energy consumption, and optimal energy consumption for stationary randomized control policy, respectively.

Proof: Because $\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1} \subset \boldsymbol{\Lambda}$, it can be shown using Lemma 2 that there exists a stationary and randomized policy Π_{opt}^i that achieves the following:

$$E\{E^{\Pi_{\text{opt}}^i}(T_k)\} = E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1}) \quad (28)$$

$$\sum_{i=1}^N E\{n_i^{\Pi_{\text{opt}}^i} a(T_k)\} = E\{n_{\text{opt}} a(T_k)\} + \sum_{i=1}^N E\{\Delta n_i^{\Pi_{\text{opt}}^i} a(T_k)\} = \sum_{i=1}^N E\{\mu_i^{\Pi_{\text{opt}}^i}(T_k)\} - \varepsilon \quad (29)$$

where $E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1})$ is the optimal energy consumption corresponding to the rate vector $\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1} \subset \boldsymbol{\Lambda}$. Substituting (28) and (29) into (6), we get:

$$\Delta_T(T_k) + VE\{E^{\text{CSSD}}(T_k) | \mathbf{Q}(T_k)\} \leq B + VE\{E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1}) | \mathbf{Q}(T_k)\} - (\varepsilon - d_{\text{total}}) E\{\sum_{i=1}^N Q_i(T_k) | \mathbf{Q}(T_k)\} \quad (30)$$

Substituting (5) into (30), and taking the expectation over $\mathbf{Q}(T_k)$, we obtain:

$$E\{L(T_k + 1) - L(T_k)\} + (\varepsilon - d_{\text{total}}) E\{\sum_{i=1}^N Q_i(T_k)\} \leq B + VE\{E_{\text{av}}^*(\boldsymbol{\lambda}^* + \mathbf{d} + \varepsilon \mathbf{1})\} \quad (31)$$

Summing (31) from $T_k = 0$ to $K - 1$, using the fact that $E^*(\lambda^* + \mathbf{d} + \varepsilon \mathbf{1}) \leq E_{\max}$,

$\sum_{i=1}^N Q_i(T_k) \leq Q_{\max}$, we have:

$$\varepsilon E\left\{\sum_{T_k=0}^K \sum_{i=1}^N Q_i(T_k)\right\} \leq KB + VKE\{E_{\max}\} + Kd_{\text{total}}Q_{\max} \quad (32)$$

Dividing both sides of (32) by εK , we obtain:

$$\frac{1}{K} \sum_{T_k=1}^K E\left\{\sum_{i=1}^N Q_i(T_k)\right\} \leq \frac{B + d_{\text{total}}Q_{\max} + VE_{\max}}{\varepsilon} \quad (33)$$

Taking the limit superior as $K \rightarrow \infty$, (26) is proven.

Moreover, based on (30), we can derive:

$$VE\{E^{\text{CSSD}}(T_k) | \mathbf{Q}(T_k)\} \leq B + VE\{E_{\text{av}}^*(\lambda^* + \mathbf{d} + \varepsilon \mathbf{1}) | \mathbf{Q}(T_k)\} + d_{\text{total}}E\left\{\sum_{i=1}^N Q_i(T_k) | \mathbf{Q}(T_k)\right\} \quad (34)$$

Taking the expectation of (34) over $\mathbf{Q}(T_k)$, then summing $T_k = 1$ to K , we have:

$$VE\left\{\sum_{T_k=1}^K E^{\text{CSSD}}(T_k)\right\} \leq KB + KVE\{E_{\text{av}}^*(\lambda^* + \mathbf{d} + \varepsilon \mathbf{1})\} + Kd_{\text{total}}Q_{\max} \quad (35)$$

Dividing both sides of (35) by KV , we obtain:

$$\frac{1}{K} E\left\{\sum_{T_k=1}^K E^{\text{CSSD}}(T_k)\right\} \leq \frac{B + d_{\text{total}}Q_{\max}}{V} + E\{E_{\text{av}}^*(\lambda^* + \mathbf{d} + \varepsilon \mathbf{1})\} \quad (36)$$

Taking the limit superior as $K \rightarrow \infty$, using the Lebesgue's dominated convergence theorem, and then letting $\varepsilon \rightarrow \infty$, we obtain inequality (27).

Based on Theorem 4, the CSSD algorithm can achieve a balance between energy consumption and queue backlog when there is a deviation between the subscribed services and optimal subscription solution. According to [21], we have $E\{E_{\text{av}}^*(\lambda^* + \mathbf{d})\} > E\{E_{\text{av}}^*(\lambda^*)\}$. By comparing (27) with (14), we can see that when $\Delta n_{\text{total}}(T_k) > 0$, we need to set V to a larger value to obtain the same time-averaged energy consumption as the case in which the subscribed services equal the optimal subscription solution, i.e., $n_{\text{real}}(T_k) = n_{\text{opt}}$. However, the time-averaged queue backlog will increase if V is increased. From these results, we can safely draw the conclusion that for the same V , the time-averaged energy consumption and queue backlog of the case where $\Delta n_{\text{total}}(T_k) > 0$ is higher than that of the case where $n_{\text{real}}(T_k) = n_{\text{opt}}$.

5. Performance Evaluation

In order to evaluate the effectiveness of our proposed CSSD algorithm in the MCC environment, we compare our algorithm with EEC algorithm. We consider the MCC environment as shown in Fig. 1. There is one cloud server provider support users over randomly varying channel. Using Java, a mobile cloud computing simulation platform is built to simulate this MCC environment. This platform mainly includes two parts, i.e. mobile Internet and cloud computing. In the cloud computing part, we employ the modeling of cloud computing infrastructure and services offered by CloudSim [22]. For the mobile Internet part, we develop a mobile Internet model reference the LET network which is described in

[19]. To make the comparison more convenient, we adopt mobile multimedia applications to represent the cloud services. The services that are offered by the cloud data center need to be delivered to the mobile users by a base station in a cell. The users walk randomly without leaving their cell. The MCC system executes the applications and delivers services data to users when the users subscribe services on the cloud. We suppose all mobile applications have the same data arrival rate, and set $\lambda=100$ Kbit/s. The CSSD algorithm can improve performance by guiding the users to subscribe to services reasonably based on the time-varying channel state and queue backlog. Without loss of generality, we suppose that the service rate of the channel is exponentially distributed. This service rate is generated from the ranges $[0, 1,000]$ where the average service rate is set to 500 Kbit/s. According to [19], the power of MDs in active and idle states are set to 10 and 0.1 mW, respectively. The time unit T is set to 1 s, the slot τ is set to 0.1 s, and parameter V is set to 10,000.

As an executor, the mobile agent operates in unit time. In our experiments, the mobile agent first detects the queue backlog and channel state of each user at the the beginning of every time unit. Next, using the queue backlog and channel state of each user, the mobile agent calculates the optimal number of service subscriptions for the current time unit. Finally, the mobile agent guides the users to subscribe services and transmits data to users in the proper slots during the current time unit. We would like to emphasize that each execution of the algorithm calculates the optimal number of service subscriptions and transmits data to users for only one time unit, not for all time units. We ran each simulation for 200 s, which is long enough to observe the performance of both the CSSD and EEC algorithms.

5.1 Network Traffic

In this experiment, we investigated the network traffic of our CSSD algorithm and the EEC algorithm. We consider scenarios in which a mobile user uses mobile applications with a time-varying channel. We let the mobile user subscribe to five mobile applications at a time under the EEC algorithm, while they subscribe mobile applications in strict accordance with the optimal subscription solution n_{opt} under the CSSD algorithm. We measured the service data arrival and service data transfer for each time unit as well as for the whole experiment. The experimental results are shown in Figs. 3 and 4, respectively.

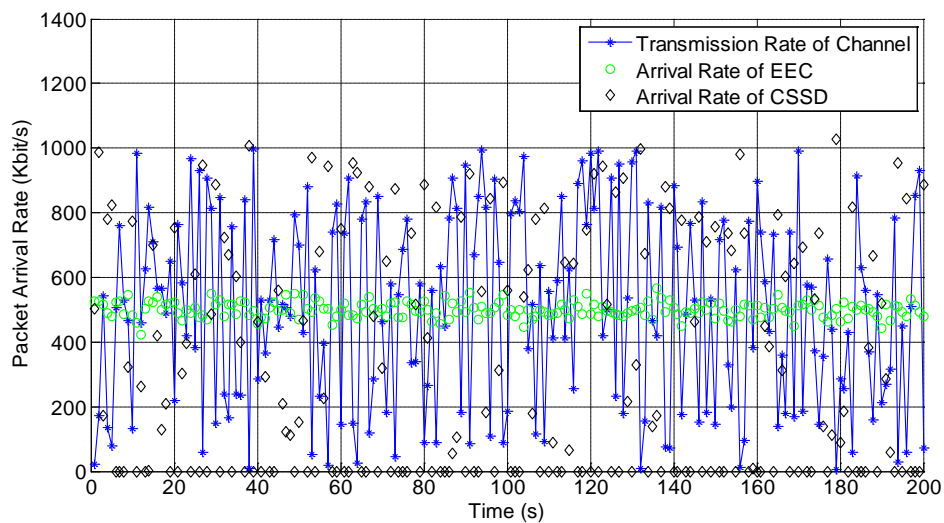


Fig. 3. Service data arrival rate comparison

Fig. 3 compares the service data arrival rate. It can be seen that the curve of our CSSD algorithm fluctuates with the changes of the channel state. Moreover, there are no service data arriving at the channel in some time units. This supports the validity of CSSD algorithm Step 2. We can also see that the curve of the EEC algorithm fluctuates around 500 Kbit/s no matter how the channel state changes. The reason for this is that the mobile user subscribes to a fixed number of mobile applications under the EEC algorithm.

Fig. 4 shows the service data transfer rate of the EEC algorithm and our CSSD algorithm. We can see that when the channel transfers service data to users, the CSSD algorithm has a higher service rate than the EEC algorithm. We also can observe that the service data transfer rates of our CSSD algorithm are more than 400 Kbit/s. The reason for this is that the EEC algorithm aims at service data delivery and balances energy efficiency and queue backlog. In contrast, our CSSD algorithm combines service subscription and delivery based on the EEC algorithm at every time unit. In this way, our CSSD algorithm has a lower queue backlog than that of EEC algorithm, as validated in **Fig. 5**. Thus, our CSSD algorithm can defer data transfer until the channel's transfer rate is high enough to support users at a high service rate. This implies that the CSSD algorithm achieves a better performance than the EEC algorithm if the MCC system has a larger queue backlog or poor channel.

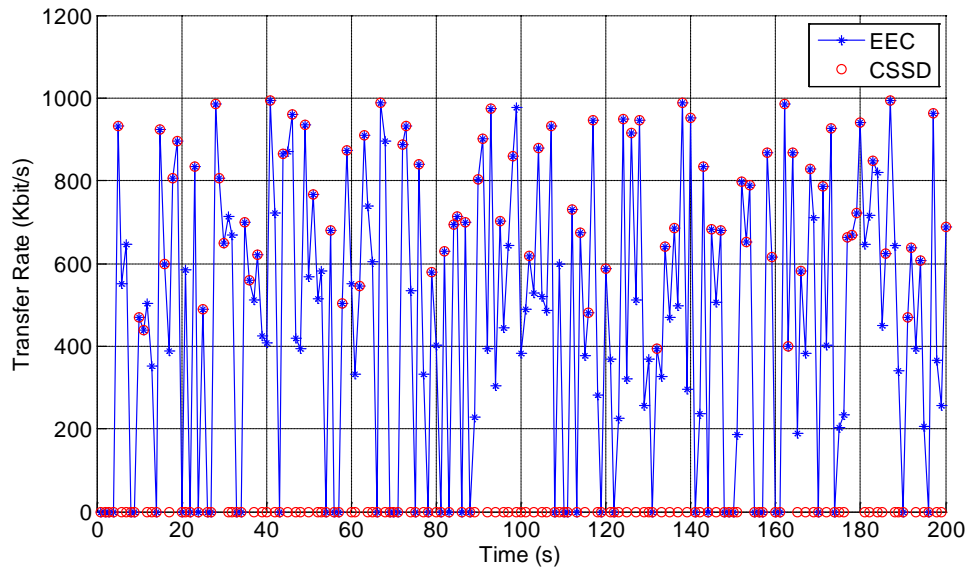


Fig. 4. Service data transfer rate comparison

5.2 Performance of Optimal Service Subscriptions

To evaluate the performance improvement of our CSSD algorithm, we consider two cases: $N=1$ and $N=2$. Let each mobile user subscribe to two mobile applications at one time under the EEC algorithm, while the mobile user subscribes to mobile applications strictly according to the optimal subscription solution n_{opt} under the CSSD algorithm. In this experiment, we use the queue backlog, energy consumption, and service subscription cost as the performance metrics to evaluate performance. We set the cost of a mobile application to 0.3 cents per time unit. This corresponds to the fixed-price model used in Microsoft's Windows Azure Platform [23].

We first examine the queue backlog of the EEC algorithm and our CSSD algorithm, as shown in Fig. 5. From this graph, our CSSD algorithm works well and the queue backlog is sufficiently small. In other words, the queue backlog of the CSSD algorithm is lower than 5,000 Kbit. For the EEC algorithm, although queue stability can be satisfied, the queue backlog of the EEC algorithm is greater than that of our CSSD algorithm. This can be regarded in the following way: when the queue backlog becomes very large, our CSSD algorithm can reduce the queue backlog by reducing the number of service subscriptions and increasing the number of transmission slots while the EEC algorithm reduces the queue backlog simply by increasing the number of transmission slots.

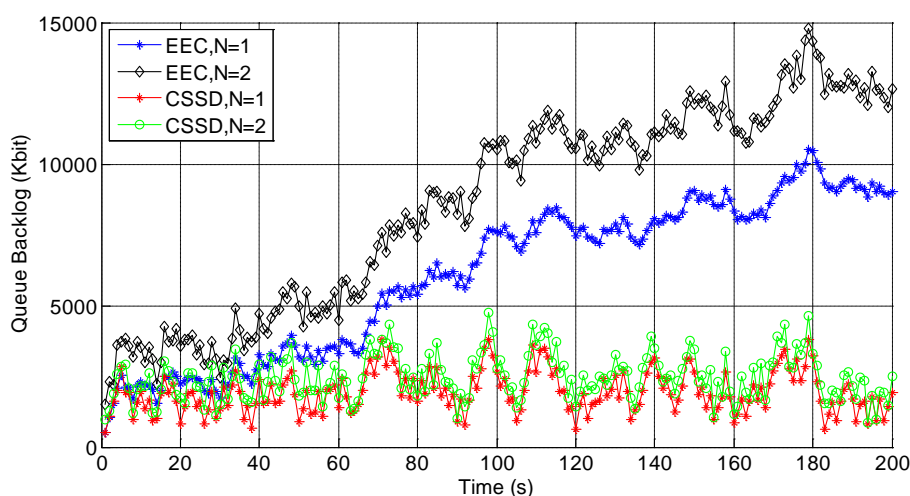


Fig. 5. Queue backlog comparison

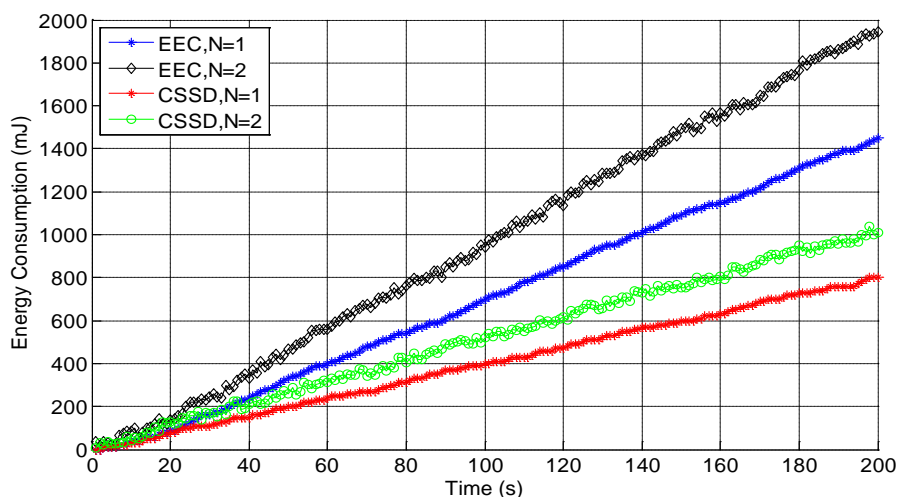


Fig. 6. Energy consumption comparison

We next examine the energy consumption of the EEC algorithm and our CSSD algorithm, as shown in Fig. 6. Compared with the EEC algorithm for $N=1$ and $N=2$, our CSSD algorithm achieves 38% and 50% lower energy consumption, respectively. This is because our CSSD algorithm can reduce the number of service subscriptions when the channel condition is poor or the queue backlog is high. In this way, our CSSD algorithm can sustain a lower queue

backlog. This result confirms the benchmark performance, as shown in Fig. 5. Thus, for the same parameter V , which control the balance between energy consumption and queue backlog, our CSSD algorithm has a better performance in terms of reducing the energy consumption.

Finally, we simulated two algorithms to compare the cost of service subscription. For ease of comparison, we define a performance metrics C_{total} that evaluates the total cost of subscribing to services:

$$C_{\text{total}} = \sum_{i=1}^N \sum_{T_k=1}^K n_i(T_k) p \quad (37)$$

Fig. 7 compares the values of C_{total} under the two algorithms for various values of N . It is clear that our proposed algorithm has a better performance than the EEC algorithm. In particular, compared with the EEC algorithm, our CSSD algorithm reduces the cost of subscribing services by 50% for $N=2$. This is because our CSSD algorithm can guide the users to subscription services according to the optimal subscription solution n_{opt} , and hence users can reduce the subscription of unnecessary services. It is interesting to note that the CSSD algorithm for $N=1$ and $N=2$ have almost same cost of service subscription. This is because users subscribe to mobile applications in strict accordance with the optimal subscription solution n_{opt} , thus the cost of service subscription does not rapidly increase with N .

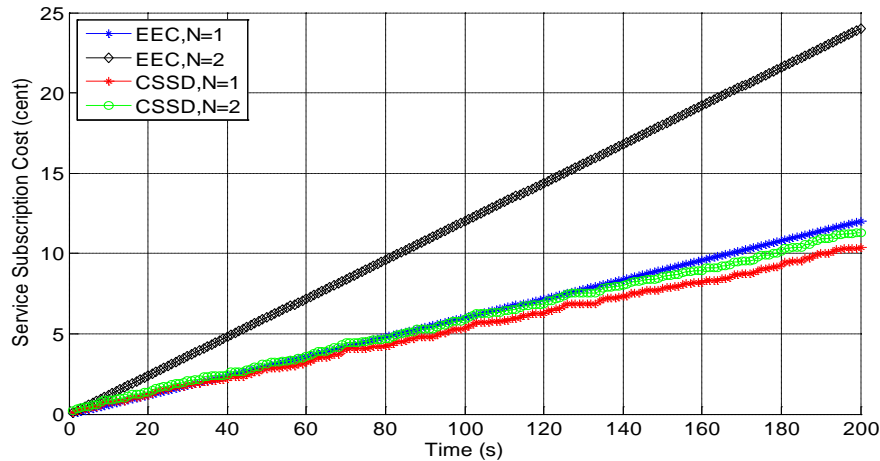


Fig. 7. Service subscription cost comparison

5.3 Impact of Service Subscribed Deviation

The purpose of this experiment was to investigate the impact of the CSSD algorithm on performance incurred by the service subscription deviation. In this experiment, we considered three different cases of expected value of service subscription deviation Δn_{total} : $E[\Delta n_{\text{total}}] = 0$, $E[\Delta n_{\text{total}}] = 0.2$, and $E[\Delta n_{\text{total}}] = 0.4$, where $E[\Delta n_{\text{total}}] = 0$ denotes the subscribed services equal to the optimal subscription solution, therefore there is no service subscription deviation. Let the number of users N be fixed to two. The experimental results are shown in Figs. 8 and 9.

Fig. 8 shows the queue backlogs of $E[\Delta n_{\text{total}}] = 0$, $E[\Delta n_{\text{total}}] = 0.2$, and $E[\Delta n_{\text{total}}] = 0.4$. It can be seen that the queue backlogs of $E[\Delta n_{\text{total}}] = 0$, $E[\Delta n_{\text{total}}] = 0.2$, and $E[\Delta n_{\text{total}}] = 0.4$ stabilize at around 1,500, 3,000, and 5,000 Kbit when the simulation time is more than 100 s, respectively. These results confirm the claim that our CSSD algorithm can consume more energy for transmission while satisfying queue stability. We also observe that the queue backlog increases with the expected value of service subscription deviation when the queue is stable. For example, the queue backlog of $E[\Delta n_{\text{total}}] = 0.4$ is the maximum. The reason for this is that the data packet arrival rate increases with the expected value of service subscription deviation, thus $E[\Delta n_{\text{total}}] = 0.4$ incurs the highest queue backlog.

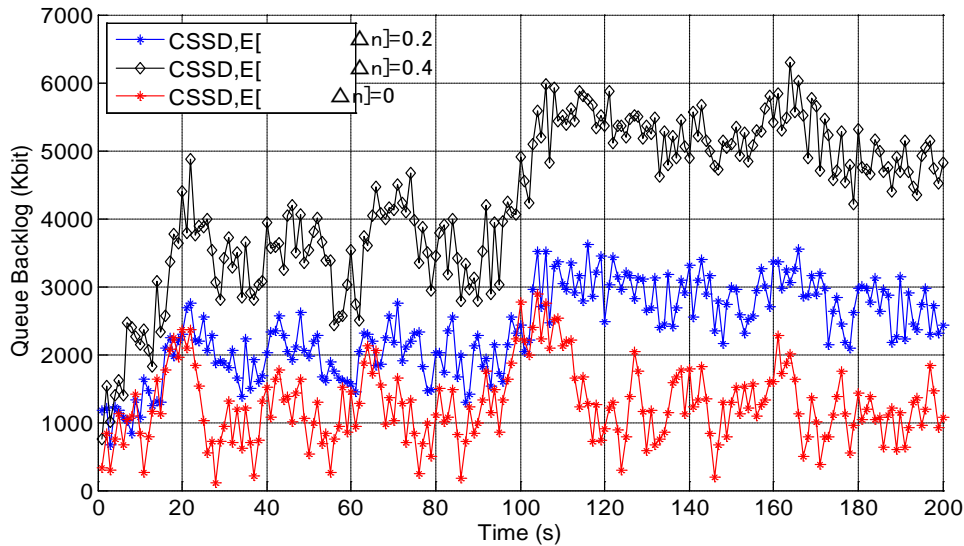


Fig. 8. Queue backlog comparison under service subscription deviation

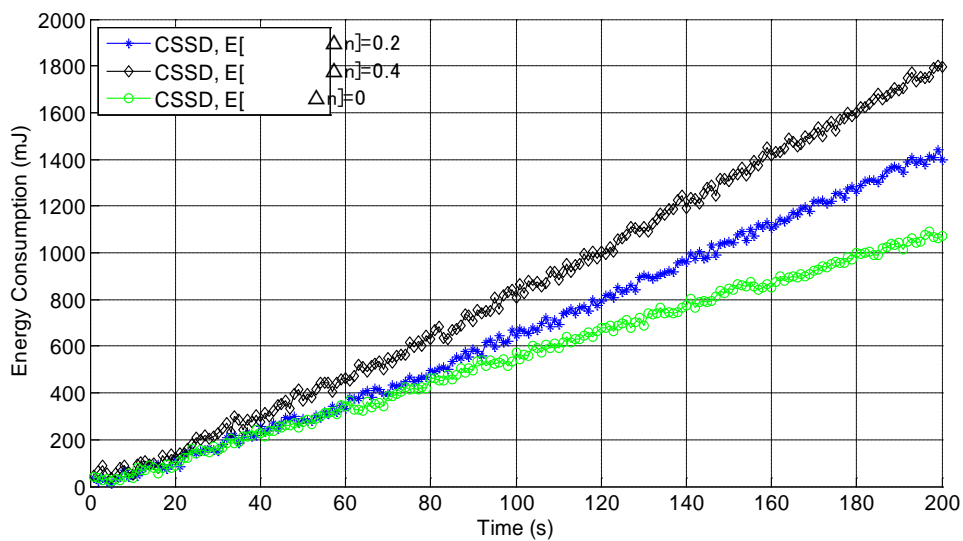


Fig. 9. Energy consumption comparison under service subscription deviation

Fig. 9 compares the energy consumption of $E[\Delta n_{\text{total}}]=0$, $E[\Delta n_{\text{total}}]=0.2$, and $E[\Delta n_{\text{total}}]=0.4$. We observe that at the beginning of the simulation, these three cases have similarly low queue backlogs and there are no data packet transfers to mobile users with poor channel conditions, thus these three cases have almost the same energy consumption. Taken as a whole, the energy consumption increases with the expected value of service subscription deviation. This is because higher expected values of service subscription deviation have higher queue backlogs. In order to maintain queue stability, data packets might be transmitted in poor channel conditions, and hence a higher expected value of service subscription deviation consumes higher energy.

6. Conclusion

In this paper, we proposed a new energy-efficient scheduling method for a time-varying channel scenario in MCC networks. This new method employs a mobile agent's ability to calculate the optimal subscription and promote cloud services to users. Based on the Lyapunov optimization, the proposed CSSD algorithm makes transmission decisions and guides users to subscribe to cloud services considering both the channel condition and queue backlog. Simulation results validate that our proposed CSSD algorithm performs better in terms of queue backlog, energy consumption, and service subscription cost.

From the simulated results, it is shown that our proposed CSSD algorithm can be used as a reference for MCC system design, providing practical engineering insights. In future work, the performance of our CSSD algorithm will be evaluated in practical mobile applications, such as mobile office, mobile commerce, mobile learning, and so on. Moreover, the more general case will be considered, where the information interaction and delay constraint bring little application performance penalty. In addition, compatibility issues with existing protocols can also be incorporated into the proposed CSSD algorithm to broaden its application.

References

- [1] Y.-H. Zhang, X.-F. Chen, J. Li, H. Li and F.-H. Li, "Attribute-Based Data Sharing with Flexible and Direct Revocation in Cloud Computing," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 11, pp. 4028 - 4049, 2014. [Article \(CrossRef Link\)](#)
- [2] F.-M. Liu, P. Shu, H. Jin, L.-J. Ding, J. Yu, D. Niu and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 14 - 22, 2013. [Article \(CrossRef Link\)](#)
- [3] M. S. Hasan, E.-N. Huh, "Heuristic based Energy-aware Resource Allocation by Dynamic Consolidation of Virtual Machines in Cloud Data Center," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 8, pp. 1825 - 1842, 2013. [Article \(CrossRef Link\)](#)
- [4] C. Ge, Z. Sun and N. Wang, "A Survey of Power-Saving Techniques on Data Centers and Content Delivery Networks," *IEEE Communication Surveys & Tutorials*, vol. 15, no. 3 pp. 1334-1354, 2013. [Article \(CrossRef Link\)](#)
- [5] D. Jung, T. Suh, H. Yu and J. M. Gil, "A Workflow Scheduling Technique Using Genetic Algorithm in Spot Instance-Based Cloud," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 9, pp. 3126 - 3145, 2014. [Article \(CrossRef Link\)](#)
- [6] N. D. Han, Y. Chung, and M. Jo, "Green Data Centers for Cloud-assisted Mobile Ad-hoc Networks in 5G," *IEEE Network*, vol.29, no.2, pp. 70-76, 2015. [Article \(CrossRef Link\)](#)
- [7] N. Shrestha, J. H. Youn, and N. Sharma, "A code-based sleep and wakeup scheduling protocol for low duty cycle sensor networks," in *Proc. of of the Int. Conf. on Networking and Information Technology*, pp. 80-85, 2010. [Article \(CrossRef Link\)](#)

- [8] S.-B. Jang and Y.-G. Kim, "Power saving and delay reduction for supporting WLAN-based fixed-mobile convergence service in smartphone," *IEEE Transactions Consumer Electron*, vol. 56 no. 4, pp. 2747-2755, 2010. [Article \(CrossRef Link\)](#)
- [9] X. Zhang, Y. Xia and S.-Y. Luo, "Energy-aware Management in Wireless Body Area Network System," *KSH Transactions on Internet and Information Systems*, vol. 7, no. 5, pp. 949 - 966, 2013. [Article \(CrossRef Link\)](#)
- [10] H.-H. Choi and J.-R. LEE, "Analysis of Energy-Delay Trade-Off for Power-Saving Mechanism Specific to Request-and-Response-Based Applications," *IEICE Transactions Communications*, vol. 97 no. 7, pp. 1422-1428, 2014. [Article \(CrossRef Link\)](#)
- [11] C.-P. Li, M. J. Neely, "Energy-Optimal Scheduling with Dynamic Channel Acquisition in Wireless Downlinks," *IEEE Transactions Mobile Computer*, vol. 9, no. 4, pp. 527-539, 2010. [Article \(CrossRef Link\)](#)
- [12] M. Zafer, E. Modiano, "Minimum energy transmission over a wireless channel with deadline and power constraints," *IEEE Transactions Automation Control*, vol. 54, no. 12, pp. 2841-2852, 2009. [Article \(CrossRef Link\)](#)
- [13] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions Information Theory*, vol. 52, no. 7, pp. 2915-2934, 2006. [Article \(CrossRef Link\)](#)
- [14] M. R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proc. of the 10th Int. Conf. on Mobile systems, applications, and services*, pp. 255-270, 2010. [Article \(CrossRef Link\)](#)
- [15] X.-F. Wang, T. T. Kwon, Y. Choi, H.-Y. Wang, and J.-C. Liu, "Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 72-79, 2013. [Article \(CrossRef Link\)](#)
- [16] X.-F. Wang and M. Chen, "PreFeed: Cloud-Based Content Prefetching of Feed Subscriptions for Mobile Users," *IEEE Systems Journal*, vol. 8, no. 1, pp. 202-207, 2014. [Article \(CrossRef Link\)](#)
- [17] M. Mishra, A. Das, P. Kulkarni and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34-40, 2012. [Article \(CrossRef Link\)](#)
- [18] M. S. Kwang, "Agent-Based Cloud Computing," *IEEE Transactions. Services Computing*, vol. 5, no. 4, pp. 564-577, 2012. [Article \(CrossRef Link\)](#)
- [19] J.-X. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *Proc. of the 10th Int. Conf. on Mobile systems, applications, and services*, PP. 225-238, 2012. [Article \(CrossRef Link\)](#)
- [20] P. Shu, F.-M. Liu, H. Jin, et al. "eTime: energy-efficient transmission between cloud and mobile devices," in *Proc. of IEEE Int. Conf. on INFOCOM*, PP. 195-199, 2013. [Article \(CrossRef Link\)](#)
- [21] M. Neely, "Delay-based network utility maximization," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 41-54, 2013. [Article \(CrossRef Link\)](#)
- [22] C.-W. Tsai, W.-C. Huang, M.-H. Chiang, M.-C. Chiang, C.-S. Yang, "A Hyper-Heuristic Scheduling Algorithm for Cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 236-250, 2014. [Article \(CrossRef Link\)](#)
- [23] S. K. Garg, S. Venugopal, J. Broberg, and R. Buyya, "Double auction-inspired meta-scheduling of parallel applications on global grids," *Journal of Parallel and Distributed Computing*, vol. 73, no. 4, pp. 450-464, 2013. [Article \(CrossRef Link\)](#)



Xing Liu received the B.E degree in Electronics and Information Engineering from Hunan Institute of Science and Technology in 2007 and received the M.S. degree in communication and information system from Chongqing University of Post and Telecommunications in 2010. He is currently pursuing the Ph.D. degree in Beijing University of Post and Telecommunications. His research interests include resource management and energy efficiency in Mobile Cloud Computing.



Chaowei Yuan received Ph.D. degree from Xi'An JiaoTong University in 1994. He is currently professor and Ph.D director the School of Information and communication Engineering of Beijing University of Posts and Telecommunications. His current research interests include the key technologies in future wireless communication, the theory of the modern signal processing and its applications and network testing.



Enda Peng received the B.E. degree in computer science and computing from Beijing University of Post and Telecommunications in 2013. He is currently pursuing the M.S. degree in Beijing University of Post and Telecommunications. His research interests include resource management and energy efficiency in Mobile Cloud Computing.



Zhen Yan received the B.S. degree in mathematics from Xiangtan University in 1997 and the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2007. He is an associate professor in the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, Beijing University of Posts and Telecommunications. His current research focuses on multimedia systems and networking.