A Utility-Aware Approach to Redundant Data Upload in Cooperative Mobile Cloud

Ji Wang, Xiaomin Zhu, Weidong Bao
College of Information System and Management
National University of Defense Technology
Changsha, Hunan, P. R. China
{wangji, xmzhu, wdbao}@nudt.edu.cn

Ling Liu
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
lingliu@cc.gatech.edu

***Abstract*—With the proliferation of mobile devices and the improvement of wireless communication technology, an increasing number of mobile devices are utilized for emergency management and healthcare monitoring. Redundant data upload to the cloud datacenters is gaining growing interest and attraction. One of the main challenges for redundant data upload in the cooperative mobile cloud is the optimization problem of how to provide high utility and high energy efficiency for data upload in the presence of intermittent connectivity and unpredictable bandwidth of wireless and mobile network. In this paper, we formulate the problem of redundant data upload in the cooperative mobile cloud as an energy-constrained utility maximization problem that aims at maximizing the amount of effective data uploaded under the energy consumption constraints. We propose an online distributed approach to enabling mobile devices to optimally make upload decisions without depending on the current state information of other devices and the prior knowledge of its own future context. We provide a rigorous theoretical analysis and an extensive suite of simulation experiments to demonstrate the effectiveness and superiority of our approach.**

***Keywords*-Mobile Cloud; Data Upload; Online Distributed Optimization; Distributed Correlated Scheduling**

I. Introduction

Recent years have witnessed a drastic increase of mobile devices. Cisco predicts that mobile users will surpass 4.5 billion in 2016 [1]. With the proliferation of mobile devices and the improvement of information technology, mobile devices are revolutionizing many aspects of our lives. To compensate the limited battery capacity and computation power of mobile devices, mobile cloud computing leverages the unlimited resource in the cloud backend for supporting resource-intensive applications running on mobile devices. The capacities of mobile devices, such as the storage capacity, are augmented by offloading applications and data to remote cloud datacenters.

Challenges of Offloading Mobile Data. A noteworthy phenomenon is that more and more mobile devices are applied to the emergency managements [2], e.g., disaster response [3], [4] and military operation [5]. Equipped with built-in sensors like cameras and microphones, mobile devices are able to collect and process a rich set of images and videos that record the scene of disaster areas. However, the network in emergent environments is highly dynamic where the intermittent connectivity and unpredictable traffic congestion make the frequent connection and offloading to remote datacenters infeasible. In such scenarios, cooperative mobile cloud [6] consisting of a group of adjacent mobile devices that share partial amounts of computation capacities with other mobile peers paves an effective avenue towards augmenting the computation capacity of mobile device.

State of Art in Cooperative Mobile Cloud. Most of the research projects on the cooperative mobile cloud have been concentrated on the problem of augmenting mobile devices' storage capacity [7], [8]. The large volume of data such as images and videos generated by one device can be offloaded to nearby devices to alleviate the storage burden of mobile devices in the presence of intermittent connectivity to the remote cloud datacenters. Thanks to the high bandwidth short-range communication and location proximity, leveraging nearby peer mobile devices can significantly reduce the cost and improve the performance compared with offloading data to remote datacenters directly. To handle the unstable wireless connection and the unpredictable mobility of mobile devices, redundant data storage becomes a popular mechanism for the cooperative mobile cloud [9], [10] where multiple duplicates of data are stored in several devices to guarantee the reliability of data, especially in emergency management applications.

However, little work to date has studied the problem of uploading data with redundancy in the cooperative mobile cloud. For the applications that require reliable distributed data storage and upload in a dynamic network, such as in the scenarios of disaster response and exploration in mountains and deserts, on one hand, the cooperative mobile leverages peer mobile devices to store large volume of data such as images and videos; and on the other hand, the cooperative mobile needs to upload data to remote cloud backend for further data mining and backup. When the dynamic upload channel is in a poor connectivity state, the data are usually duplicated with several copies stored in the peer mobile devices. After some interval of time, each mobile device in the cooperative mobile cloud decides whether or not to upload its data. How to make an efficient upload decision is a critical and open problem for the cooperative mobile cloud. There are several obstacles to addressing this problem. *First*, in the adverse environment of disaster response and military operation, using a centralized solution to making the upload decision for each mobile device may incur unacceptable communication and computation cost. Also, the centralized approach can incur single-point failure problem. *Second*, with the unpredictable mobility and the dynamic multi-hop network between mobile devices, it is hard for one mobile device to know the state and decision of other mobile devices in a timely fashion. Two or more devices may upload the duplicates of the same data, which brings no benefit to the cooperative mobile cloud while wastes the energy of mobile devices in performing redundant data upload. *Finally*, the unstable network makes the device context highly dynamic and unpredictable. It is impossible to solve the upload problem precisely with an offline optimization approach in reality.

In this paper, we focus on the problem of uploading data with redundant duplicates in the cooperative mobile cloud with the objective of maximizing upload utility. The cooperative mobile cloud is formed by a group of mobile devices sharing their storage capacity to store data generated by others cooperatively. In order to guarantee the reliable storage, the same data may have multiple duplicates stored among these mobile devices. Facing the upload challenges discussed above, we design an online distributed scheduling algorithm based on the idea of *distributed correlated optimization approach* [11], which enables each mobile device to make an independent upload decision without the prior knowledge of mobile devices' future context. The main contributions of this work are summarized as follows: *First*, to the best of our literature knowledge, this is the first work towards the efficient decision of uploading data with redundant duplicates in the cooperative mobile cloud by formulating the problem as an energy-constrained utility maximization problem. *Second*, an online distributed optimization framework is proposed in this paper to help each mobile device make an upload decision independently. The rigorous theoretical analysis argues that the proposed algorithm is arbitrary close to the optimum over the long run. *Finally*, extensive simulation experiments are conducted to demonstrate the effectiveness of our proposed optimization framework.

The rest of the paper is organized as follows. Section II gives a brief discussion on the related work and Section III describes the problem formulation. We describe the design of our online distributed optimization framework in Section IV. Section V verifies our algorithm through a series of simulation experiments. We conclude the paper in Section VI.

II. Related Work

The storage augmentation in cooperative mobile cloud is a hot topic of both industry and academia. Moon *et al*. [12] proposed an energy efficient storage augmentation approach in mobile ad hoc network. A service-oriented framework is designed by Abolfazli *et al*. [7] to alleviate storage limitations by sharing resources of nearby mobile devices. In order to guarantee the reliability of data, the redundant data storage is widely used. Phoenix [8], a distributed storage protocol, ensures data reliability by maintaining at least two copies of data in the mobile cloud for one-hop network. Chen *et al*. [10] presented a k-out-of-n framework that is able to retrieve and process data successfully as long as k out of n mobile devices are available. The previous work enables the storage augmentation by the usage of adjacent mobile devices. Nonetheless, the problem concerning how to upload these data in the cooperative mobile cloud is still unsolved.

Uploading data to remote cloud datacenters incurs a heavy energy burden on mobile devices, especially in a poor upload channel. Many researches strive to address the issue of energy-efficient data upload in a dynamic network. Lombardo *et al*. [13] exploited the Markov model to design an energy-efficient transmission protocol that is adaptive to the uplink channel state. Xiang *et al*. [14] formulated the data transmission problem as a discrete-time stochastic dynamic program. By solving this problem, both the data throughput and the energy consumption can be optimized. Fang *et al*. [15] presented an online algorithm based on the Lyapunov optimization theory for optimally controlling the data transmission. The algorithm is able to minimize the energy cost and the dropping penalty without any prior knowledge of channel state. These existing researches focus on the energy-saving problem for one single mobile device. In this paper, we attempt to optimize the correlated upload procedure for multiple mobile devices in a dynamic network.

Some existing researches in the field of mobile sensing try to solve the problem of distributed data collection [16], [17]. Unfortunately, most of these researches ignore the effect of redundant data which bring no utility while still incur cost [16]. Besides, the upload channel state is not considered in these researches [17]. Differing from the mobile sensing problem, our work focuses on the energy-efficient data transmission of mobile devices. The channel state has a considerable effect on the system performance. We attempt to optimize the redundant data upload in a dynamic network, and hence enable mobile devices to upload data in a highly efficient manner.

III. Problem Formulation

A cooperative mobile cloud consisting of N mobile devices is considered in our work. The cooperative mobile cloud conducts upload procedures repeatedly. We use t to denote the t-th upload procedure of the cooperative mobile cloud. During the period between two upload procedures, some mobile devices generate large volume of data such as images and videos that record the scene of disaster area. To augment the storage capacity of one single mobile device and improve the reliability of data, these data are divided into several data segments with the same size which are then duplicated to several copies stored among the distributed mobile devices. $d_i(t) = (d_{i1}(t), d_{i2}(t), ..., d_{iK}(t))$ represents whether a duplicate of data segment is stored in the i-th mobile device at the time of t-th upload procedure, where K is the type of different data segments. $d_{ik}(t) = 1$ means that the duplicate of k-th data segment is stored in the i-th mobile device, and $d_{ik}(t) = 0$ otherwise. Let $\boldsymbol{d}(t)$ be a vector of these $d_i(t)$, $\boldsymbol{d}(t) = (d_1(t), d_2(t), ..., d_N(t))$.

When the upload channel is available, the cooperative mobile cloud conducts a upload procedure. Considering the different location of mobile devices, the channel states of different devices may vary. Let $\omega_i(t)$ denote the channel state of i-th mobile device in the t-th upload procedure, where $\omega_i(t) \in \Omega = \{0, 1, ..., |\Omega| - 1\}$. A large value of $\omega_i(t)$

means that the i-th mobile device is in a good channel state. Hence, $\omega_i(t) = |\Omega| - 1$ indicates the best channel state while $\omega_i(t) = 0$ is the poorest. Let $\boldsymbol{\omega}(t)$ be a vector of these $\omega_i(t)$, $\boldsymbol{\omega}(t) = (\omega_1(t), \omega_2(t), ..., \omega_N(t))$.

In the t-th upload procedure, mobile devices make decisions on whether or not to upload the stored data segments based on their own observed states. A binary variable $\alpha_i(t) \in \{0, 1\}$ is used to denote the decision of i-th mobile device. $\alpha_i(t) = 1$ indicates that the i-th mobile device decides to upload its stored data segments in the t-th upload procedure, and $\alpha_i(t) = 0$ otherwise. Let $\boldsymbol{\alpha}(t)$ be a vector of these $\alpha_i(t)$, $\boldsymbol{\alpha}(t) = (\alpha_1(t), \alpha_2(t), ..., \alpha_N(t))$. Then, the amount of effective data segments (i.e., utility) collected in the t-th upload procedure can be calculated as follows:

$$u(t) = \widehat{u}(\boldsymbol{\alpha}(t), \boldsymbol{d}(t)) = \sum_{j=1}^{K} \min\{\sum_{i=1}^{N} d_{ij}(t)\alpha_i(t), 1\} \tag{1}$$

The above utility function is derived from the fact that each data segment has multiple duplicates stored in several mobile devices. When two or more devices upload the duplicates of the same data segment, only one duplicate is effective while the others are redundant. We assume that a mobile device uploads all the data in every upload procedure if it decides to upload data. Therefore, the energy consumption can be determined by the channel state. Let $p_i(t)$ be the energy consumption of i-th device in the t-th upload procedure. We have the following equation:

$$p_i(t) = \widehat{p}(\alpha_i(t), \omega_i(t)) \tag{2}$$

The energy consumption function $\widehat{p}(\alpha_i(t), \omega_i(t))$ is studied sufficiently in the empirical transmission energy models of the previous researches [18]. Hence, we do not give the details in this section. However, no matter what the detailed form of energy consumption function is, it is obvious that a poorer channel state incurs higher energy consumption of mobile devices if they decide to upload data. Recalling the definition of $\omega_i(t)$, it can be concluded that $\widehat{p}(\alpha_i(t), \omega_i(t))$ is non-increasing with the variable $\omega_i(t)$.

The main objective of cooperative mobile cloud is to maximize the average amount of effective data segments collected over the repeated upload procedures. Nonetheless, the energy consumed by the uploads should be constrained by certain values when taking the limited battery capacity of mobile devices into consideration. Hence, we give the following energy-constrained utility maximization problem:

$$\max: \quad \overline{u} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{t-1} \mathbb{E}[u(t)] \tag{3}$$

$$\text{s.t.}: \quad \overline{p_i} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{t-1} \mathbb{E}[p_i(t)] \leq c_i, \forall i \tag{4}$$

$$\text{Decisions are distributed.} \tag{5}$$

where c_i is the average energy cost of i-th mobile device.

IV. Online Distributed Optimization Framework

It is a challenging work to solve the given energy-constrained utility maximization problem in practice. Due to the mobility of devices and the unstable upload channel, the device context is highly dynamic and unpredictable, which makes it impossible to solve the problem precisely with an offline optimization approach in reality. What makes matters worse is that a mobiles device does not know the device context and the decision of others when it decides whether or not to upload its data. In such a distributed environment, a mobile device may upload redundant data segments that bring no increment of utility while still consume energy. In order to overcome these difficulties, we exploit the advantage of *distributed correlated optimization approach* [11] to design an online distributed scheduling algorithm that enables each mobile device to make an optimal decision independently.

A. Complexity Pruning

In a upload procedure, each mobile device observes its channel state and decides whether or not to upload its data. We define $\alpha_i = \widehat{\alpha_i}(\omega_i)$ where the i-th mobile device makes a decision α_i as a deterministic function of ω_i. The distributed decision of mobile devices is defined as a vector of $\widehat{\alpha_i}(\omega_i)$:

$$\widehat{\boldsymbol{\alpha}}(\boldsymbol{\omega}) = (\widehat{\alpha}_1(\omega_1), \widehat{\alpha}_2(\omega_2), ..., \widehat{\alpha}_N(\omega_N))$$

For each mobile device, it has two possible decisions, $\widehat{\alpha_i}(\omega_i) = 0, 1$, in a channel state ω_i. Consequently, the total number M of possible strategies $\widehat{\boldsymbol{\alpha}}(\boldsymbol{\omega})$ is $M = \prod_{i=1}^{N} 2^{|\Omega|}$. $\widehat{\boldsymbol{\alpha}}^{(m)}(\boldsymbol{\omega})$ for $m \in \{1, ..., M\}$ represents a specified distributed strategy among the M possible strategies. In the given problem, we strive to choose a strategy $\widehat{\boldsymbol{\alpha}}^{(opt)}(\boldsymbol{\omega})$ among the possible strategies to maximize the amount of effective data segments collected. Nonetheless, in reality, the exponential value of M can be extremely large, which makes it infeasible to enumerate possible strategies. Fortunately, based on our analysis, most possible strategies are noneffective. It is able to prune the value of M to a polynomial size. The detailed complexity pruning is shown in the following part.

Theorem 1: For the problem given by (3)-(5), the optimal strategy $\widehat{\alpha_i}(\omega_i)$ is non-decreasing with the variable ω_i, for all $i \in \{1, ..., N\}$.

Proof: Fix two channel state $\omega, \gamma \in \Omega$, and $\omega < \gamma$. Suppose the optimal strategy satisfies $\widehat{\alpha_i}(\omega) > \widehat{\alpha_i}(\gamma)$. We proof the theorem by finding new strategies that are able to satisfy the non-decreasing property without loss of optimality.

Because $\widehat{\alpha_i}(\omega) > \widehat{\alpha_i}(\gamma)$, we have $\widehat{\alpha_i}(\omega) = 1$, $\widehat{\alpha_i}(\gamma) = 0$. Give two new strategies as below:

$$\widehat{\alpha_i}^{low}(\omega_i) = \begin{cases} \widehat{\alpha_i}(\omega_i) & \text{if } \omega_i \notin \{\omega, \gamma\} \\ 0 & \text{if } \omega_i \in \{\omega, \gamma\} \end{cases}$$

$$\widehat{\alpha_i}^{high}(\omega_i) = \begin{cases} \widehat{\alpha_i}(\omega_i) & \text{if } \omega_i \notin \{\omega, \gamma\} \\ 1 & \text{if } \omega_i \in \{\omega, \gamma\} \end{cases}$$

The above two strategies both satisfy the non-decreasing property. Suppose the i-th mobile device is in the channel state ω, γ with the probability $pr_i(\omega)$ and $pr_i(\gamma)$, respectively. Then we define a new stochastic strategy $\widehat{\alpha_i}'(\omega_i)$ as:

- $\widehat{\alpha}_i^{low}(\omega_i)$ with probability $pr_i(\gamma)/(pr_i(\omega)+pr_i(\gamma))$;
- $\widehat{\alpha}_i^{high}(\omega_i)$ with probability $pr_i(\omega)/(pr_i(\omega)+pr_i(\gamma))$.

We use $[\alpha_i, \boldsymbol{\alpha}_i^-]$ to denote the N-dimensional vector $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}_i^-$ represents the $(N-1)$-dimensional vector of α_j, $\forall j \neq i$. Then, the utility function and the energy consumptions of i-th device can be calculated as below:

- If $\omega_i(t) = \omega$, and $\widehat{\alpha}_i^{low}(\omega_i)$ is chosen as the new strategy, then $u(t) = \widehat{u}([1, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))$, $u'(t) = \widehat{u}([0, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))$; $p_i(t) = \widehat{p}(1, \omega)$, $p_i{}'(t) = \widehat{p}(0, \omega)$;
- If $\omega_i(t) = \gamma$, and $\widehat{\alpha}_i^{high}(\omega_i)$ is chosen as the new strategy, then $u(t) = \widehat{u}([0, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))$, $u'(t) = \widehat{u}([1, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))$; $p_i(t) = \widehat{p}(0, \gamma)$, $p_i{}'(t) = \widehat{p}(1, \gamma)$;
- If neither of the above two conditions are held, then $u(t) = u'(t)$, $p_i(t) = p_i{}'(t)$.

Then, we have:

$$\begin{aligned}
\mathbb{E}[u(t) - u'(t)] = & \, pr_i(\omega) \frac{pr_i(\gamma)}{pr_i(\omega) + pr_i(\gamma)} \cdot \\
& (\widehat{u}([1, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t)) - \widehat{u}([0, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))) \\
& + pr_i(\gamma) \frac{pr_i(\omega)}{pr_i(\omega) + pr_i(\gamma)} \cdot \\
& (\widehat{u}([0, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t)) - \widehat{u}([1, \boldsymbol{\alpha}_i^-(t)], \boldsymbol{d}(t))) \\
& = 0
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[p_i(t) - p_i{}'(t)] = & \, pr_i(\omega) \frac{pr_i(\gamma)}{pr_i(\omega) + pr_i(\gamma)} \cdot (\widehat{p}(1, \omega) - \widehat{p}(0, \omega)) \\
& + pr_i(\gamma) \frac{pr_i(\omega)}{pr_i(\omega) + pr_i(\gamma)} \cdot (\widehat{p}(0, \gamma) - \widehat{p}(1, \gamma))
\end{aligned}$$

When $\alpha_i(t) = 0$, the energy consumed by uploading data is zero. Hence, $\widehat{p}(0, \omega) = \widehat{p}(0, \gamma) = 0$. In addition, considering $\widehat{p}(\cdot)$ is non-increasing with $\omega_i(t)$, we have $\widehat{p}(1, \omega) \geq \widehat{p}(1, \gamma)$. Therefore, $\mathbb{E}[p_i(t) - p_i{}'(t)] \geq 0$.

Consequently, the new strategy not only satisfies the non-decreasing property but also reduces the averaged energy consumption $\mathbb{E}[p_i(t)]$ without loss of $\mathbb{E}[u(t)]$. Because the other energy consumptions $p_j(t)$, $\forall j \neq i$ are not determined by $\alpha_i(t)$, their values remain the same. Therefore, we are able to conclude that the new strategy satisfies the non-decreasing property without loss of optimality. ■

Theorem 1 argues that the optimal strategy $\widehat{\alpha}_i(\omega_i)$ tends to be large when ω_i is large. Hence, the optimal strategy has the following form:

$$\widehat{\alpha}_i(\omega_i(t)) = \begin{cases} 0 & \text{if } \omega_i(t) < \omega_i^*(t) \\ 1 & \text{if } \omega_i(t) \geq \omega_i^*(t) \end{cases}$$

Consequently, the problem is transformed into finding the threshold ω_i^* for each mobile device. As a mobile device has $|\Omega|$ possible thresholds, the number of effective strategies is pruned to $\widetilde{M} = \prod_1^N |\Omega|$, which is acceptable because the number of mobile devices participating the cooperative mobile cloud in one region is usually not quite large.

B. Online Distributed Scheduling Algorithm

The key idea of distributed correlated optimization approach is that each mobile device chooses a strategy among $\{\widehat{\boldsymbol{\alpha}}^{(1)}(\boldsymbol{\omega}), \widehat{\boldsymbol{\alpha}}^{(2)}(\boldsymbol{\omega}), ..., \widehat{\boldsymbol{\alpha}}^{(\widetilde{M})}(\boldsymbol{\omega})\}$ in each upload procedure. It is assumed that all mobile devices receive feedback concerning the channel states $\boldsymbol{\omega}(t)$ and the decisions $\boldsymbol{\alpha}(t)$ at the end of $(t+D)$-th upload procedure, where $D \geq 0$ represents the feedback delay of the system. This assumption is feasible in reality, and can be easily implemented by piggybacking.

To begin with, we first transform the energy cost constraints (4) into a queue stability problem. For each mobile device, define a virtual queue $Q_i(t)$, and $\mathbf{Q}(t) = (Q_1(t), ..., Q_N(t))$. $Q_i(t)$ is updated by the following equation at the end of t-th upload procedure:

$$Q_i(t+1) = \max\{Q_i(t) + p_i(t-D) - c_i, 0\} \tag{6}$$

where $Q_i(0) = 0$, $p_i(-1) = ... = p_i(-D) = 0$. Each mobile device maintains the queues $\mathbf{Q}(t)$ and updates the queues based on the feedback at the end of upload procedure repeatedly. According to the proof in [19], if the virtual queues are stable (i.e., $\lim_{t\to\infty} \mathbb{E}[Q_i(t)/t] = 0$, $\forall i$), then the energy cost constraints (4) are satisfied. We define the Lyapunov function as below:

$$L(t) = \frac{1}{2} \sum_{i=1}^{N} Q_i(t)^2 \tag{7}$$

The Lyapunov function represents a scalar metric of queue congestion of $\mathbf{Q}(t)$. A small value of $L(t)$ indicates that the virtual queues of mobile devices have strong stability. In another word, the energy cost constraints are satisfied. To push the Lyapunov function away from a congestion state, we then define the D-slot Lyapunov drift:

$$\Delta(t+D) = L(t+D+1) - L(t+D) \tag{8}$$

Intuitionally, minimizing the above D-slot conditional Lyapunov drift is able to keep queue stability. To maximize the utility function (1), the drift-plus-penalty technique [19] is introduced, which transforms the energy-constrained utility maximization problem into the minimization of the upper bound for the following expression in each upload procedure:

$$\mathbb{E}[\Delta(t+D) - Vu(t)|\mathbf{Q}(t)] \tag{9}$$

The parameter $V(\geq 0)$ is designed to control the tradeoff between the queue stability and the utility. A lower value of V implies that the cooperative mobile cloud tends to maintain the queue stable (i.e., lower energy consumption) rather than collect more effective data segments. The following lemma provides the upper bound of expression (9).

Lemma 1: For a given $V > 0$. One has for each upload procedure:

$$\mathbb{E}[\Delta(t+D) - Vu(t)|\mathbf{Q}(t)] \leq A(1+2D) - \sum_{i=1}^{N} c_i Q_i(t) + \mathbb{E}[\sum_{i=1}^{N} p_i(t) Q_i(t) - Vu(t)|\mathbf{Q}(t)] \tag{10}$$

where $A = \frac{\sum_{i=1}^{N} c_i^2}{2}$.

Instead of minimizing the drift-plus-penalty expression directly, our approach strives to minimize the right-hand-side (RHS) of inequation (10), and thus to maximize the lower bound of the utility function while guaranteeing the stability of $\mathbf{Q}(t)$; that is to say, the constraint (4) is satisfied. As a result, by introducing the drift-plus-penalty technique, the energy-constrained utility maximization problem can be solved by the approach that each mobile device chooses a strategy among $\{\widehat{\boldsymbol{\alpha}}^{(1)}(\boldsymbol{\omega}), \widehat{\boldsymbol{\alpha}}^{(2)}(\boldsymbol{\omega}), ..., \widehat{\boldsymbol{\alpha}}^{(\widetilde{M})}(\boldsymbol{\omega})\}$ to minimize the RHS of inequation (10) in each upload procedure.

Because the mobile devices do not know the channel states and decisions of other devices in the current upload procedure, the mobile device is unable to calculate $p_i(t)$ and $u(t)$ in the RHS of inequation (10). However, the feedback mechanism makes the information concerning the channel states $\boldsymbol{\omega}(t-D)$ and the decisions $\boldsymbol{\alpha}(t-D)$ available at the end of t-th upload procedure. Motivated by the method proposed in [20], $p_i^{(m)}(t)$ and $u^{(m)}(t)$ when using strategy $\widehat{\boldsymbol{\alpha}}^{(m)}(\boldsymbol{\omega})$ can be approximated by:

$$\widetilde{p}_i^{(m)}(t) = \frac{1}{S}\sum_{s=1}^{S} \widehat{p}_i(\widehat{\alpha}_i^{(m)}(\omega_i(t-D-s)), \omega_i(t-D-s))$$

$$\widetilde{u}^{(m)}(t) = \frac{1}{S}\sum_{s=1}^{S} \widehat{u}(\widehat{\boldsymbol{\alpha}}^{(m)}(\boldsymbol{\omega}(t-D-s)), \boldsymbol{d}(t-D-s))$$

where S is a positive integer representing a sample size.

The pseudocode of the online distributed scheduling algorithm of each mobile device is presented in Algorithm 1.

Algorithm 1: Algorithm in t-th Upload Prodecure

```
foreach mobile device i ∈ {1, ..., N} do
    Observe the channel state ω_i(t) and the queue Q(t);
    Choose a strategy among {α̂^(1)(ω), ..., α̂^(M̃)(ω)} that
    minimizes Σ_{i=1}^{N} p̃_i^(m)(t) Q_i(t) − V ũ^(m)(t);
    Conduct the action α_i(t) = α̂_i^(m)(ω_i(t));
    Receive the feedback concerning ω(t − D) and α(t − D),
    and update the queue Q(t) based on (6);
```

C. Performance Analysis

The following Theorem 2 gives the performance gap between the optimal solution and the solution achieved by our algorithm.

Theorem 2: For arbitrary channel states of mobile devices, for any $V \geq 0$ and any $S \geq 0$, we have:

- The gap between the optimal utility and the utility achieved by our algorithm is:

$$\overline{u}^{opt} - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[u(t)] \leq \frac{A(1+2D)}{V} + \frac{\mathbb{E}[L(D)]}{VT} + O(1/\sqrt{S}), \tag{11}$$

  where $\overline{u}^{opt}$ is the maximum averaged utility under the constraints of energy cost.
- Our proposed algorithm guarantees that the averaged energy cost of each mobile device satisfy:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[p_i(t)] \leq c_i + O(\sqrt{\frac{V}{T}}),\ \forall i \tag{12}$$

Proof: In each upload procedure, the drift-plus-penalty technique makes a decision to minimize the RHS of inequation (10). Using the optimal solution to the problem (3)-(5) that can be achieved by the exact correlated scheduling [11], we have:

$$\mathbb{E}[\Delta(t+D) - Vu(t)|\mathbf{Q}(t)] \leq A(1+2D) - V\overline{u}^{opt}$$

Taking expectations of the above inequation, we have:

$$\mathbb{E}[\Delta(t+D) - V\mathbb{E}[u(t)]] \leq A(1+2D) - V\overline{u}^{opt}$$

Summing over $t \in \{0, ..., T-1\}$, we can get:

$$\begin{aligned} &\mathbb{E}[L(T+D)] - \mathbb{E}[L(D)] - V\sum_{t=0}^{T-1}\mathbb{E}[u(t)] \\ &\leq AT(1+2D) - VT\overline{u}^{opt} \end{aligned} \tag{13}$$

Because $\mathbb{E}[L(T+D)] \geq 0$, rearranging the above inequation, we have:

$$\overline{u}^{opt} - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[u(t)] \leq \frac{A(1+2D)}{V} + \frac{\mathbb{E}[L(D)]}{VT}$$

Our algorithm uses the delayed feedback to approximate the exact value. According to the analysis in [20], the performance gap between the approximate method and the exact correlated scheduling is $O(1/\sqrt{S})$. Therefore, the inequation (11) holds.

Rearranging inequation (13) again, we have:

$$\mathbb{E}[L(T+D)] \leq (B+CV)T$$

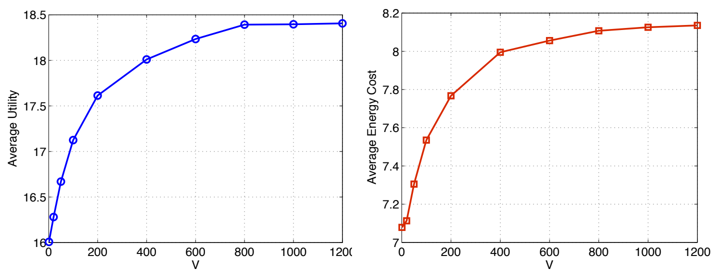
where $B = \mathbb{E}[L(D)] + A(1+2D)$, and C is defined as a constant satisfying $C \geq \mathbb{E}[u(t)] - \overline{u}^{opt}$

Recalling the definition of $L(t)$, we have:

$$\mathbb{E}[\sum_{i=1}^{N} Q_i(T+D)^2] \leq 2(B+CV)T$$

By the Jensen's inequation:

$$\frac{\mathbb{E}[\sum_{i=1}^{N} Q_i(T+D)]}{T} \leq \sqrt{\frac{2(B+CV)}{T}}$$



(a) Utility vs. V (b) Energy Cost vs. V

Fig. 1: Cost-Utility Tradeoffs

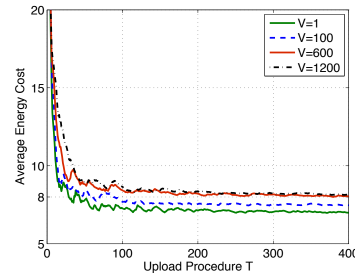


Fig. 2: Average Energy Cost with Different V

According to the proof in [19]:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[p_i(t)] \leq c_i + \frac{\mathbb{E}[Q_i(T+D)]}{T}$$

Combining the above two inequations, we can conclude that the inequation (12) holds. ■

Theorem 2 provides the gap between the averaged utility achieved by our algorithm and the optimal utility as well as the upper bounds on the averaged energy cost. Inequation (11) indicates that by choosing a sufficient large value of V, the averaged utility can be pushed arbitrarily close to the optimal value. While a too large value of V, as shown in inequality (12), will incur a higher energy cost. The cooperative mobile cloud is able to make a flexible tradeoff between the utility and the energy cost by adjusting the parameter V. Besides, a larger sample size S also can push the solution to the optimal value. However, it, at the same time, causes a longer scheduling time and larger space to store the states.

V. Performance Evaluation

In order to evaluate the performance of our proposed framework, we conduct a series of simulation experiments in this section. A cooperative mobile cloud consisting of 8 mobile devices is simulated in our experiments. There are at most 20 types of different data segments generated between two upload procedures. To guarantee the data reliability, each data segment is duplicated into 3 copies stored in these mobile devices. There are 4 types of upload channel states denoted by $\Omega = \{0, 1, 2, 3\}$. In each upload procedure, the channel state of a mobile device is randomly chosen from Ω with the same probability. According to the empirical transmission energy models [18], the transmission energy is inversely proportional to the channel state ω_i. Hence, the upload energy consumption for a mobile device in different channel states is set as {60,30,12,6} (J), respectively. The average energy cost constraint of each mobile device is set as 8J. The default feedback delay is $D = 2$, and the sample size is $S = 20$.

A. Verification of Cost-Utility Tradeoffs

We first verify the effectiveness of parameter V in our algorithm as a control parameter to make a tradeoff between the utility and the energy cost. Fig. 1(a) demonstrates that the utility improves and converges to the optimal value with the increase of V. However, the increase diminishes as the utility approaches the optimal value gradually. This experimental result verifies Theorem 2 which argues that the average utility can be pushed to the optimal value with a gap of $O(1/V)$. Nonetheless, the improvement of utility adversely aggravates the energy burden of mobile devices as shown in Fig. 1(b). Fortunately, our algorithm effectively avoids an excessive energy cost of mobile devices. When $V < 600$, the average energy cost is fewer than the constraint $c = 8$J. Even when $V = 1200$, the average energy cost is merely 1.67% larger than the constraint. Together, the results show the $[O(1/V), O(\sqrt{V})]$ tradeoff between the utility and the energy cost, which coincides with Theorem 2. To achieve a large utility under the given energy cost constraint, we set $V = 600$ in the following experiments.

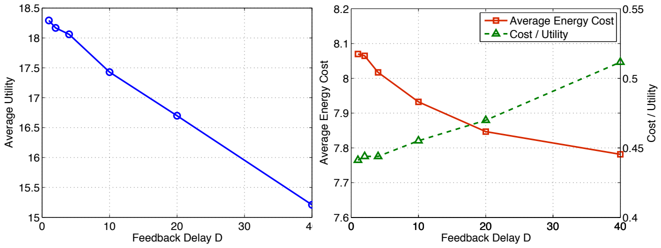
We then further verify whether the energy cost constraint is satisfied with different values of V. It can be seen from Fig. 2 that the average energy cost with different values of V descends significantly and converges quickly to the energy cost constraint. For a smaller value of V, the decent speed is higher. The average energy cost with $V = 1, 100$ rapidly descends below the constraint after several upload procedures. Meanwhile, the cost with $V = 600, 1200$ also converges to the constraint when $t > 100$. These results argue that larger values of V achieve a higher utility while suffer from prolonged time for average energy cost to converge to the constraint.

B. Impact of Feedback Delay

Fig. 3 shows the performance impact of parameter D. It can be found from Fig. 3(a) that the value of parameter D has a significant impact on the utility. When the feedback delay D is prolonged, the average utility descends obviously. It is intuitive to conclude that shortening the feedback delay of the cooperative mobile cloud can notably improve its performance. Thanks to the piggybacking technique, the feedback information usually can be delivered in a relatively short time. Although Fig. 3(b) shows the decrease of energy cost, the curve of energy efficiency (Cost/Utility) indicates that the system suffers a deteriorated energy efficiency when the value of D becomes large, which means that the cooperative mobile cloud consumes more energy to upload data.

C. Impact of Channel State

To illustrate the impact of channel state and the adaption of our algorithm, we conduct a series of experiments where the probability distribution of channel states change. The upload



(a) Utility vs. D (b) Energy Cost vs. D

Fig. 3: Impact of parameter D

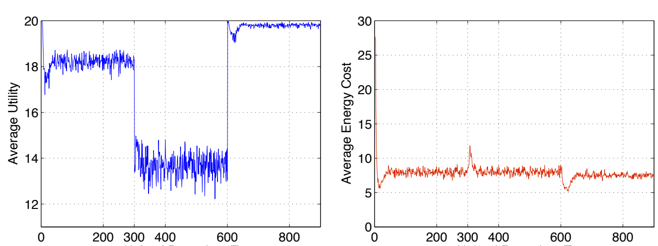
TABLE I: Distribution of Channel State

$Pr[\omega_i(t)]$	$\omega_i(t) = 0$	$\omega_i(t) = 1$	$\omega_i(t) = 2$	$\omega_i(t) = 3$
Type 1	0.45	0.45	0.05	0.05
Type 2	0.05	0.05	0.45	0.45

procedure is increased to 900 times that is divided into three phases. In the phase 1 ($t < 300$), the channel states for all mobile devices have the same probabilities as given in the previous experiments. In the phase 2 ($300 \le t < 600$), the probability distributions abruptly changes to Type 1 distribution (a poor channel state) as shown in Table I. In the phase 3 ($t \ge 600$), the channel states are randomly chosen with Type 2 distribution (a good channel state).

Fig. 4 demonstrates the average utility and the average energy cost over the 900 upload procedures. Values at each upload procedure are averaged over 100 independent simulation runs. The two vertical lines in Fig. 4(a) indicate that the cooperative mobile cloud can adjust to the new optimal value once the channel states change. This result argues that our algorithm is adaptive to the unpredictable environment change. In addition, the average utility achieved in the phase 2 is much lower than that in the phase 1, based on which we can conclude that the channel state is a bottleneck for improving the system utility. Fig. 4(b) plots the average energy cost versus upload procedures. Two noticeable disturbances can be observed at the changes of channel states distribution. Nonetheless, the energy cost quickly re-converges to the constraint after a few upload procedures.

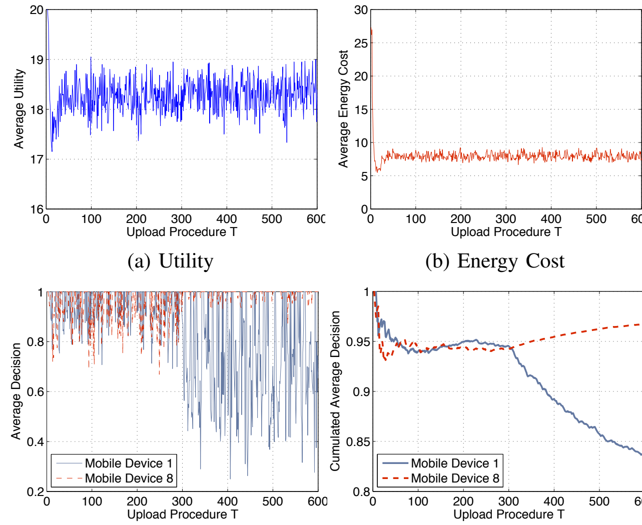
Considering the differences of channel states among mobile devices, we further verify the adaption of our algorithm to the abrupt changes of one mobile device. 600 upload procedures



(a) Utility (b) Energy Cost

Fig. 4: Impact of Channel State

are divided into two phases. In the phase 1 ($t < 300$), the channel states for all mobile devices are uniformly distributed over Ω. In the phase 2 ($t \ge 300$), the probability distribution of mobile device 1 changes to Type 1 distribution, and that of mobile device 8 changes to Type 2 distribution. Other mobile devices keep the same probability distribution.



(a) Utility (b) Energy Cost

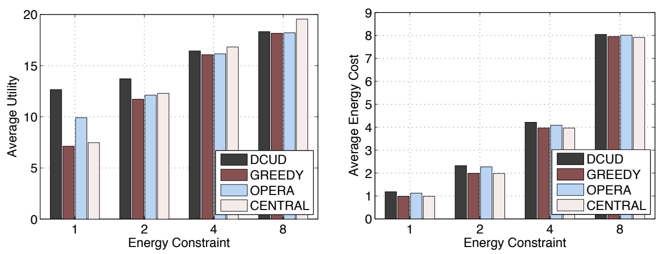
(c) Average Decision over 100 runs (d) Cumulated Average Decision

Fig. 5: Impact of Independent Changes

From Fig. 5(a) and (b), it can be observed that neither the average utility nor the average energy cost shows an obvious disturbance at the change of channel state distribution. In order to investigate the reason for this phenomenon, we plot the average decisions of the mobile device 1 and 8. The decisions $\alpha_1(t)$ and $\alpha_8(t)$ at each upload procedure are averaged over 100 independent runs. Recalling the definition of $\alpha_i(t)$, a large value of average decision means that the mobile device tends to upload data at this upload procedure. In the phase 1, the average decision of device 1 is similar to that of mobile device 8. However, in the phase 2, as the channel state of mobile device 1 becomes poor, its average decision fluctuates significantly, and drops below 0.5 at many upload procedures. On the contrary, the average decision of device 8 keeps above 0.95 in the phase 2, which indicates that the device 8 almost decides to upload data at each upload procedure over the 100 runs. To show the trend clearly, we plot the cumulated average decision $\sum_{\tau=0}^{\tau=t} \overline{\alpha}_i(\tau)$ versus upload procedures in Fig. 5(d). The curve of device 1 shows an obvious descending trend after $t = 300$, while the curve of device 8 ascends. These experimental results demonstrate the excellent adaption of our algorithm. Although the mobile devices have no knowledge about the current states of other devices when they make decisions, relying on the advantages of distributed correlated optimization, our algorithm still enables the distributed devices to work collaboratively.

D. Performance Comparison

In order to show the performance improvement of our algorithm (called **d**istributed **c**orrelated **u**pload **d**ecision, *DCUD*), we compare it with three benchmark algorithms. *GREEDY* is a distributed algorithm where a mobile device decides to upload data once the average energy cost is below the constraint. *OPERA* proposed in [15] is an online algorithm aiming at minimizing the energy consumption and the data dropping rate. Differing from these algorithms, *CENTRAL* is a centralized algorithm that derives from *GREEDY* based on the knowledge about the states of all mobile devices.



(a) Utility (b) Energy Cost

Fig. 6: Performance Comparison

Fig. 6 shows the performance comparison of the four algorithms with different energy constraints. The results demonstrate a noticeable improvement of our algorithm, especially when the energy constraint is tight, a common scenario in emergency managements. When $c_i = 1$, the utility improvement comparing with *GREEDY* is up to 77.5%. When the energy constraint is loose, our algorithm still outperforms the benchmark algorithms except for *CENTRAL*. However, *CENTRAL*, a centralized solution, is assumed to have the states information of all mobile devices, which is infeasible in the adverse environment as discussed in Section I.

VI. Conclusion

The cooperative mobile cloud is gaining a popularity as an effective mechanism for augmenting the storage capacity of mobile devices reliably through nearby mobile devices. One of the main challenges in such cooperative mobile cloud is how to upload the data with redundant copies to the remote cloud backend. This paper formulates this problem as an energy-constrained utility maximization problem. Inspired by the idea of distributed correlated optimization approach, an online distributed scheduling algorithm is developed to enable each mobile device to make an independent decision without the prior knowledge of future context. We provide a rigorous theoretical analysis on reducing the complexity of the algorithm. It is proved that our algorithm can approach an average utility that is arbitrarily close to optimum, while satisfying the energy consumption constraints. We conduct a series of experiments to demonstrate the effectiveness of our online distributed optimization approach.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under grants No. 61572511, and No. 11428101, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20134307110029, the Hunan Provincial Natural Science Foundation of China under grant No. 2015JJ3023 as well as the Southwest Electron & Telecom Technology Institute under Grant No. 2015014. Ling Liu is partially supported by the National Science Foundation under Grants IIS-0905493, CNS-1115375, NSF 1547102, SaTC 1564097, and Intel ISTC on Cloud Computing.

References

[1] I. Cisco, “Cisco visual networking index: Forecast and methodology, 2011C2016,” *CISCO White paper*, pp. 2011-2016, 2012.

[2] K. Mitra, C. Ahlund et al., “A mobile cloud computing system for emergency management,” *IEEE Cloud Computing*, vol. 1, no. 4, pp. 30-38, 2014.

[3] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, “Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response,” *Communications Magazine IEEE*, vol. 48, no. 3, pp. 128-136, 2010.

[4] Y. Kikuchi and Y. Shibata, “Mobile cloud computing for distributed disaster information system in challenged communication environment,” *in IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pp. 512-517, 2015.

[5] W. Gao, “Opportunistic peer-to-peer mobile cloud computing at the tactical edge,” *in 2014 IEEE Military Communications Conference*, 2014.

[6] M. V. Pedersen and F. H. P. Fitzek, “Mobile clouds: the new content distribution platform,” *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1400-1403, 2012.

[7] S. Abolfazli, Z. Sanaei, M. Shiraz, and A. Gani, “Momcc: marketoriented architecture for mobile cloud computing based on service oriented architecture,” *in 1st IEEE International Conference on Communications in China Workshops*, pp. 8-13, 2012.

[8] R. K. Panta, R. Jana, F. Cheng, Y.-F. R. Chen, V. A. Vaishampayan, “Phoenix: storage using an autonomous mobile infrastructure,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1863-1873, 2013.

[9] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, “Resource allocation for energy efficient k-out-of-n system in mobile ad hoc networks,” *in 22nd International Conference on Computer Communications and Networks*, pp. 1-9, 2013.

[10] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, “Energy-efficient fault-tolerant data storage and processing in mobile cloud,” *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 28-41, 2015.

[11] M. J. Neely, “Distributed stochastic optimization via correlated scheduling,” *in 2014 Proceedings IEEE INFOCOM*, pp. 2418-2426, 2014.

[12] A. Moon and H. Cho, “Energy efficient replication extended database state machine in mobile ad hoc network,” *in IADIS International Conference on Applied Computing. Citeseer*, pp. 224-228, 2004.

[13] A. Lombardo, C. Panarello, and G. Schembra, “A model-assisted crosslayer design of an energy-efficient mobile video cloud,” *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2307-2322, 2014.

[14] 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.

[15] W. Fang, X. Yin, Y. An, N. Xiong, Q. Guo, and J. Li, “Optimal scheduling for data transmission between mobile devices and cloud,” *Information Sciences*, vol. 301, no. 12, pp. 169-180, 2012.

[16] X. Sheng, J. Tang, and W. Zhang, “Energy-efficient collaborative sensing with mobile phones,” *in 2012 Proceedings IEEE INFOCOM*, pp. 1916-1924, 2012.

[17] Y. Han, Y. Zhu, and J. Yu, “Utility-maximizing data collection in crowd sensing: An optimal scheduling approach,” *in 12th IEEE International Conference on Sensing, Communication, and Networking*, pp. 345-353, 2015.

[18] 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.

[19] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1-211, 2010.

[20] M. J. Neely, S. T. Rager, and T. F. La Porta, “Max weight learning algorithms for scheduling in unknown environments,” *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1179-1191, 2012.