Extend Your Journey: Considering Signal Strength and Fluctuation in Location-based Applications

Chih-Chuan Cheng and Pi-Cheng Hsiu, *Member, IEEE*

Abstract—Reducing the communication energy is essential to facilitate the growth of emerging mobile applications. In this paper, we introduce signal strength into location-based applications to reduce the energy consumption of mobile devices for data reception. First, we model the problem of data fetch scheduling, with the objective of minimizing the energy required to fetch location-based information without impacting the application's semantics adversely. To solve the fundamental problem, we propose a dynamic-programming algorithm and prove its optimality in terms of energy savings. Then, we perform postoptimal analysis to explore the tolerance of the algorithm to signal strength fluctuations. Finally, based on the algorithm, we consider implementation issues. We have also developed a virtual tour system integrated with existing web applications to validate the practicability of the proposed concept. The results of experiments conducted based on real-world case studies are very encouraging and demonstrate the applicability of the proposed algorithm towards signal strength fluctuations.

Index Terms—Energy-efficient optimization, signal strength, signal fluctuation, cellular data fetch scheduling, location-based applications

1 Introduction

Recent years have witnessed a paradigm shift in personal computing. The popularity of mobile devices equipped with location-sensing technology has enabled the expansion of many existing information services by adding a location dimension. A variety of *location-based applications and services* have progressively permeated people's daily life, ranging from the services for directions or recommendations about nearby attractions to social interaction with friends via location sharing [23]. Location-based applications will become more diverse and pervasive due to the potential for a range of highly personalized and context-aware services [6]. However, the trend will lead to a significant boost in mobile data traffic and, consequently, result in further pressure on the limited battery capacity of mobile devices. Thus, reducing the communication energy is an imminent challenge in stimulating the development of emerging location-based applications.

Many existing approaches leverage the complementary characteristics of WiFi and 3G, i.e., WiFi to improve energy efficiency and 3G to maintain ubiquitous connectivity [1, 2, 18–20]. Recently, it has been observed that signal strength has a direct impact on the communication energy consumption. The communication energy per bit when the signal is weak could be as much as six times more than that when the signal is strong [21]. This phenomenon has proved evident in both WiFi [15] and 3G [21]. The reason for such a phenomenon results mainly from the adaptive modulation and power control employed in wireless network systems. Based on the observation, it could be promising to exploit signal strength information to reduce the communication energy of mobile devices. However, the challenge is how to exploit this observation to gain energy efficiency. In particular, signal strength may fluctuate with time due to multipath fading [10], so attention has to be paid to the impact of signal fluctuations on the practicability of the proposed approaches in real-world environments.

In this paper, our major contribution is to introduce signal strength into location-based applications to reduce the energy consumption of mobile devices for data reception. To validate the practicability of the concept, we developed a virtual tour system comprised of an on-line server and a mobile application program based on Android. First, we model the fundamental problem in the virtual tour system as a data fetch scheduling problem. Second, we propose a dynamic-programming algorithm to solve the fundamental problem. The solution involves scheduling the fetching of location-based information at appropriate locations so as to minimize the total energy consumption. We prove that the algorithm is optimal in terms of energy savings. Third, we perform postoptimal analysis to explore how the algorithm responds to signal strength fluctuations, especially the fluctuation range within which the derived solution remains optimal or feasible. The analysis helps to understand the impact of signal fluctuations on the practicability of this new concept in real-world environments. Fourth, we discuss technical implementation issues that arise when introducing signal strength into location-based applications for energy savings. Fifth, we conducted a series of experiments in Taipei City for real-world case studies. The results show that an Android smartphone of HTC EVO 3D can achieve a significant energy reduction when accessing location-based applications. Finally, we discuss the limitations of our work and highlight issues that require further investigation. The concept, once proved practicable and embraced gradually, could be extended and applied to other variants of location-based applications based on the knowledge learnt from this work.

• *The authors are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan. E-mail: {charleycheng, pchsiu}@citi.sinica.edu.tw*

A preliminary version of this paper appears in the Proceedings of the IEEE International Conference on Computer Communications (INFOCOM) 2013.

The rest of this paper is organized as follows. Section 2 describes the system model and defines the problem. In Section 3, we propose an optimal algorithm to solve the problem, which is followed by the postoptimal analysis in Section 4. In Section 5, we discuss system implementations. The experiment results are reported in Section 6. Section 7 discusses the limitations of our work and possible extensions. Section 8 reviews related work, and Section 9 contains some concluding remarks.

2 System Model and Problem Definition

Suppose that a mobile user carrying a smart mobile device is traveling from a source to a destination. Along the route, the user is provided with location-based information, such as local maps and directions, based on the geographical locations. In addition, the user may also be provided with entertainment information of interests, like travel recommendations. That is so called *social location-based services*. Recently, *augmented reality techniques* have also been introduced into location-based applications, like Layar, to further improve user experience. Such information is comprised of *objects*, e.g., map tiles, photos, and video clips. Some objects are requested by the user on demand and, thus, force the user to wait for immediate communication[1]. In contrast, some objects are supposed to be displayed along the route and fetched automatically.

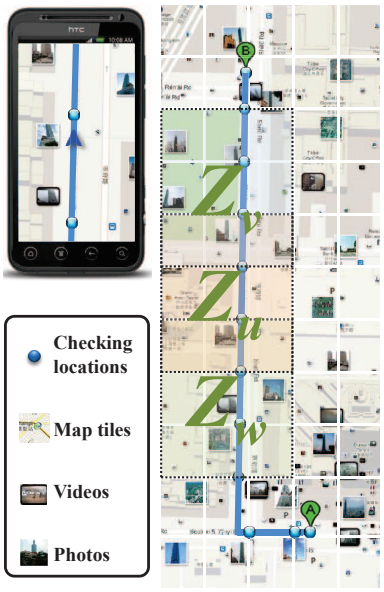


Fig. 1. An example location-based application

Let the objects to be displayed along the route be denoted by $O = \{o_1, o_2, \cdots, o_m\}$. In practical implementations, objects are usually of various sizes and fetched in an atomic manner. Let each object $o_i \in O$ be of size s_i (bytes). Moreover, to avoid the overheads incurred by maintaining which objects have been (or have not been) dispatched to certain users, objects are normally dispatched in some predefined order, referred to as the *dispatch constraint* in our scheduling problem. On the other hand, along the route exists a set of *checking locations*, $P = \{p_1, p_2, \cdots, p_n\}$, at which location-based information updates may be triggered. The checking locations are usually situated at the intersections of the route and the boundaries of map tiles, shown by the blue bullets in Figure 1. It is assumed that the user will visit the checking locations in order; otherwise, a new route is derived once the user deviates from the original route. At a checking location p_u, a *local scene*, Z_u, comprising some objects will appear on the device screen. Consequently, for the route, there is a corresponding set of local scenes $\mathbb{Z} = \{Z_1, Z_2, \cdots, Z_n\}$, where two adjacent local scenes may be overlapped. To ensure the application's semantics, the corresponding local scene should be available before the user arrives at each checking location, except the source whose local scene is deemed to be requested on demand. Thus, the *availability constraint* is imposed on our scheduling problem. Finally, the time required to download a given amount of information depends on the downlink data rate. Because the data rate has its limitation and the user will pass through a checking location in a finite time, the amount of information that can be fetched at a location is also limited. Thus, each checking location should be associated with a maximum fetch size, reflected by the *fetch constraint* in our scheduling problem.

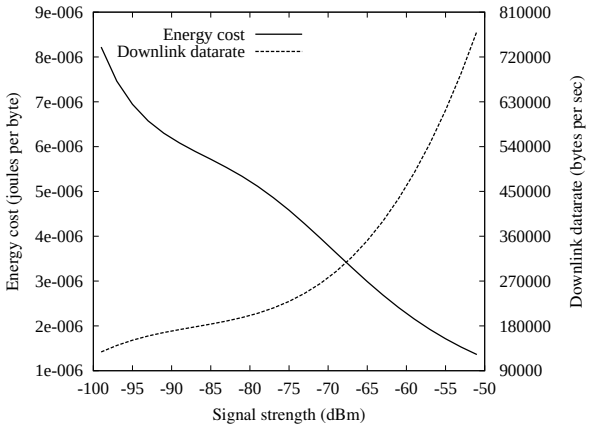


Fig. 2. Energy and data rate models

It requires time and consumes energy to fetch location-based information. It has been noticed that the data rate and the communication energy have strong relationship with signal strength. The relationship can be established with a monotonic function using regression-based techniques [24]. Figure 2 shows the downlink data rate (bytes per second) with respect to the signal strength, as well as the energy cost (joules per byte) based on an Android smartphone of HTC EVO 3D. The signal strength may vary from location to location. Nevertheless, as observed in [21], the signal strength at a location is generally stable over time, which makes it possible to measure and collect the signal strengths at the checking locations. As a result, the route is associated with a set of energy costs $E = \{e_1, e_2, \cdots, e_n\}$, each of which corresponds to a checking location, as well as a set of maximum fetch sizes $C = \{c_1, c_2, \cdots, c_n\}$ determined based

1. We do not consider the objects requested on demand because their energy consumption depends on whether, and where, the user downloads them.

on a set of data rates $D = \{d_1, d_2, \cdots, d_n\}$. In addition to the energy consumption for data communication, there is a non-ignorable energy cost, known as *tail energy* and denoted as $\tilde{e}$, after each data transfer. The estimation of the energy costs and the determination of the maximum fetch sizes will be discussed further in Section 5.

On the basis of the above descriptions, the communication energy required to fetch the same object may be different at different locations. This raises a technical problem: which objects should be fetched at which checking locations such that the total communication energy[2] is minimized, without impacting the application's semantics adversely? The solution involves scheduling the fetching of objects at appropriate locations. A mapping of a set of checking locations, P, to a set of objects, O, is called a fetch schedule $\sigma : u \in [1, n] \rightarrow \{o_i \in O\}$. A fetch schedule is *feasible* if the aforementioned scheduling constraints, namely the dispatch, availability, and fetch constraints, are satisfied. Next, we formally define the *data fetch scheduling problem* (DFSP):

Instance: A set of checking locations $P = \{p_1, p_2, \cdots, p_n\}$, where each location $p_u \in P$ is associated with an energy cost e_u, a maximum fetch size c_u, and a local scene Z_u; a set of objects $O = \{o_1, o_2, \cdots, o_m\}$, where each object $o_i \in O$ is associated with a size s_i, and a tail energy cost $\tilde{e}$ for each data transfer.

Objective: A feasible fetch schedule σ that minimizes the total energy consumption for data reception,

$$\sum_{\substack{\forall p_u \in P \\ \sigma(u) \neq \{\phi\}}} \left(\tilde{e} + \sum_{\forall o_i \in \sigma(u)} s_i \times e_u \right).$$

3 Energy-Efficient Cellular Data Fetch Schedule Optimization

This section proposes a dynamic-programming algorithm to solve the data fetch scheduling problem. First, we present the algorithm, analyze its time complexity, and prove its optimality in Section 3.1. Finally, we provide a simple example to better explain the algorithm in Section 3.2.

3.1 Algorithm Description

The basis of the proposed algorithm is the recursive formula given in Equation (1). Before discussing the formula, we define some terms. Let $\texttt{LastOt}(v)$ denote the maximum subindex of the objects in local scene Z_v, i.e., $\texttt{LastOt}(v) = \max\{j \mid o_j \in Z_v\}$. Let $\texttt{FarOt}(u, i)$ be the maximum subindex of the objects that we can fetch at p_u from o_{i+1}, i.e., $\texttt{FarOt}(u, i) = \max\{j \leq m \mid \sum_{k=i+1}^{j} s_k \leq c_u\}$. Similarly, let $\texttt{FarLn}(u, i)$ be the maximum subindex of the checking locations reachable from p_u if we fetch as many objects as possible at p_u from o_{i+1}, i.e., $\texttt{FarLn}(u, i) = \max\{v \leq n \mid \sum_{k=i+1}^{\texttt{LastOt}(v)} s_k \leq c_u\}$. The recursive formula allows us to compute $E(u, i)$, which is defined as the minimum energy required to reach p_n from p_u when the first i objects have been available on the device already. Accordingly, our objective is to compute $E(1, 0)$. We delineate the three possible cases in Equation (1):

1) If $u = n$, then $E(u, i)$ is set at 0. That is, location p_u is the destination p_n; thus, no energy is required to fetch objects to reach p_n from p_u.
2) If $u \geq \texttt{FarLn}(u, i)$, then $E(u, i)$ is set at ∞. That is, it is impossible to reach a checking location farther than p_u, even if objects are downloaded from o_{i+1} sequentially until the maximum fetch size c_u is achieved. Thus, the energy consumption is set at ∞ to indicate there is no feasible fetch schedule to reach p_n from p_u.
3) Otherwise, if $u < \texttt{FarLn}(u, i)$, there exists a checking location p_v such that $u < v \leq \texttt{FarLn}(u, i)$, as shown by the example in Figure 3. Suppose that no objects will be fetched before p_v. To ensure that p_v is reachable from p_u, local scene Z_v should be available on the device before the user leaves p_u. Thus, if object o_j is the last one fetched at p_u, then the possible values of j should be in the range from $\max(i, \texttt{LastOt}(v))$ to $\texttt{FarOt}(u, i)$. We consider all the possible values to derive a candidate solution to $E(u, i)$. If $j = i$, it means that no objects are fetched at p_u. By definition, the minimum energy required to reach p_n from p_v, with the first i objects available on the device, is $E(v, i)$. On the other hand, if $j \neq i$, the energy consumption involves the energy required to fetch objects from o_{i+1} to o_j at p_u, i.e., $\sum_{k=i+1}^{j} s_k \times e_u + \tilde{e}$, and then the minimum energy required to reach p_n from p_v, i.e., $E(v, j)$. Note that the derived solution is based on the presupposition that no objects will be fetched before p_v. To relax the presupposition, by considering all the possible values of v, $E(u, i)$ is set as the minimum of the candidate solutions derived.

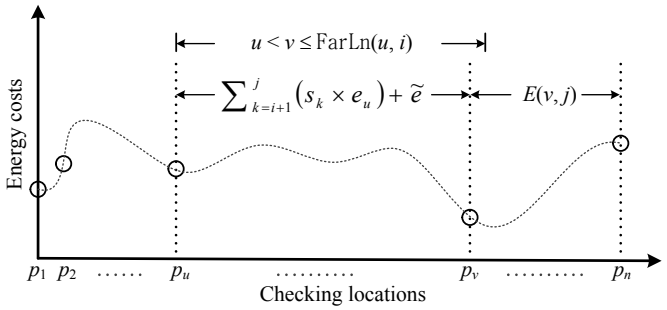


Fig. 3. An illustration to the dynamic-programming formula

The dynamic-programming formula in Equation (1) is recursively implemented by an algorithm, referred to as Algorithm 1, presented in [5] to determine a data fetch schedule. Interested readers are referred to [5]. Algorithm 1 has the following properties.

***Lemma** 1: The time complexity of Algorithm 1 is $O(m^2 n^2)$.*

Proof: See [5]. □

***Theorem** 1: Algorithm 1 is an optimal algorithm for the data fetch scheduling problem.*

Proof: See [5]. □

2. Data retransmission is not considered in the problem definition because of its uncertainty. We assess its effect in real-world experiments in Section 6.

$$E(u,i)=\begin{cases}0, & \text{if } u=n;\\ \infty, & \text{else if } u\geq \texttt{FarLn}(u,i);\\ \min\limits_{\substack{u<v\leq \texttt{FarLn}(u,i)\\ \max(i,\texttt{LastOt}(v))\leq j\leq \texttt{FarOt}(u,i)}} \left\{\begin{array}{ll} E(v,i), & \text{if } j=i;\\ E(v,j)+\tilde{e}+\sum_{k=i+1}^{j} s_k\times e_u, & \text{if } j\neq i.\end{array}\right\} & \text{otherwise.}\end{cases} \quad (1)$$

3.2 A Simple Example

In this section, we provide a simple example to better explain Algorithm 1. Consider seven objects to be displayed on a route with four checking locations, where the respective object sizes are 45, 60, 50, 40, 70, 35, and 50 Kbytes, and the local scenes of the checking locations are $\mathbb{Z}=\{\{o_1\},\{o_1,o_2\},\{o_2,o_3,o_4,o_5\},\{o_4,o_5,o_6,o_7\}\}$. Suppose that the energy costs at the four locations are $E=\{4.9\times 10^{-6}, 5.7\times 10^{-6}, 5\times 10^{-6}, 5.1\times 10^{-6}\}$ joules per byte, and the maximum fetch sizes (determined based on the data rates $D=\{30,15,25,23\}$ Kbytes per second) are $C=\{180,150,170,160\}$ Kbytes. For ease of presentation, the tail energy cost is assumed to be 0.5 joules.

Figure 4 shows the table T completed by Algorithm 1 for the simple example. Let us consider three entries, $T[1,0]$, $T[2,2]$, and $T[4,7]$, which belong to the three cases in Equation (1), respectively. After each table entry is initialized as -1, the algorithm starts to compute the solution to $E(1,0)$. Since $u=1<\texttt{FarLn}(1,0)=2$, entry $T[1,0]$ belongs to the third case (represented by Lines 11-18 in Algorithm 1). Because $1<v\leq \texttt{FarLn}(1,0)=2$, the only possible value of v is 2; furthermore, because $\max(0,\texttt{LastOt}(2))=2\leq j\leq \texttt{FarOt}(1,0)=3$, the possible values of j are 2 and 3. Thus,

$$\begin{aligned}T[1,0]&=\min\left\{\begin{array}{l}E(2,2)+0.5+105\times 2^{10}\times 4.9\times 10^{-6}\\ E(2,3)+0.5+155\times 2^{10}\times 4.9\times 10^{-6}\end{array}\right\}\\ &=\min\left\{\begin{array}{l}T[2,2]+1.027\\ T[2,3]+1.278\end{array}\right\}\\ &=\min\left\{\begin{array}{l}\infty+1.027\\ 2.077+1.278\end{array}\right\}\\ &=3.355,\end{aligned}$$

where $T[2,2]=\infty$ and $T[2,3]=2.077$ can be derived as follows. For entry $T[2,2]$, since $u=2=\texttt{FarLn}(2,2)$, it belongs to the second case and thus is set at ∞ (Lines 9-10). For entry $T[2,3]$, since $u=2<\texttt{FarLn}(2,3)=3$, it belongs to the third case and can be computed in a similar way as $T[1,0]$. The first case occurs when $u=4=n$; actually, only the last table entry, $T[4,7]$, belongs to this case and is set at 0 (Lines 7-8).

u \ i	0	1	2	3	4	5	6	7
1	3.355	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	∞	2.077	-1	-1	-1	-1
3	-1	-1	-1	-1	-1	0.935	0.756	-1
4	-1	-1	-1	-1	-1	-1	-1	0

Fig. 4. Table T computed by Algorithm 1 for the example

A corresponding feasible schedule σ can be constructed by tracing table T as follows. We begin with the table entry $T[1,0]$ and examine each entry $T[v,j]$, where $1<v\leq \texttt{FarLn}(1,0)$ and $\max(0,\texttt{LastOt}(v))\leq j\leq \texttt{FarOt}(1,0)$, until we find an entry such that the solution to $E(1,0)$ equals to 3.355 stored in $T[1,0]$. As $T[2,3]$ is such an entry, we schedule objects o_1, o_2, and o_3 to be fetched at p_1. Next, we start with entry $T[2,3]$ and find entry $T[3,5]$ that leads $E(2,3)$ to be equal to 2.077. Accordingly, we schedule objects o_4 and o_5 to be fetched at p_2. Then again, we start with entry $T[3,5]$ and find entry $T[4,7]$ that has $E(3,5)=0.935$. Thus, objects o_6 and o_7 are scheduled to be fetched at p_3. Finally, since $u=4=n$, we end with $T[4,7]$ and derive a fetch schedule $\sigma=\{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\},\varnothing\}$.

4 Postoptimal Analysis

This section performs postoptimal analysis to investigate the tolerance of Algorithm 1 to the signal strength fluctuations. First, we present the procedure of the postoptimal analysis in Section 4.1. Then, we provide a simple example to better explain the postoptimal analysis in Section 4.2. Finally, in Section 4.3, the properties that support the postoptimal analysis are proved.

4.1 Approach Description

The fetch schedule derived by Algorithm 1 is proved to be optimal with respect to the estimated signal strength. We do not expect that the estimated signal strength is exactly equal to the real signal strength. When there exists an *estimation error*, defined as the real signal strength subtracted by the estimated signal strength, we perform postoptimal analysis to investigate the tolerance of the algorithm to the estimation error. In particular, the analysis indicates (1) the fluctuation range within which the derived schedule remains optimal or feasible; (2) the difference in energy consumption between the schedules derived based on the real signal strength and the estimated signal strength; and (3) the estimation error boundaries where the optimal schedule changes.

Assume that a signal strength estimator generates a set of estimated signal strength $\Gamma=\{\varsigma_1,\varsigma_2,\cdots,\varsigma_n\}$ dBm, each of which includes an estimation error within ε dBm, where ε is a real number. That is, the real signal strength is bounded by $\Gamma_\varepsilon=\{\varsigma_1+\varepsilon,\varsigma_2+\varepsilon,\cdots,\varsigma_n+\varepsilon\}$ dBm. We define DFSP$_\varepsilon$ as the data fetch scheduling problem with respect to the real signal strength Γ_ε. For each checking location p_u, let $\Delta e_u(\varepsilon)$ and $\Delta c_u(\varepsilon)$ denote, respectively, the differences in the energy cost and in the maximum fetch size between the real signal strength and the estimated signal strength. That is, the energy costs and the maximum fetch sizes with respect to the real signal strength are $E_\varepsilon=\{e_1+\Delta e_1(\varepsilon),e_2+\Delta e_2(\varepsilon),\cdots,e_n+\Delta e_n(\varepsilon)\}$ and $C_\varepsilon=\{c_1+\Delta c_1(\varepsilon),c_2+\Delta c_2(\varepsilon),\cdots,c_n+\Delta c_n(\varepsilon)\}$. To reflect

the change incurred by the estimation error ε, the terms $\texttt{FarOt}(u,i)$ and $\texttt{FarLn}(u,i)$ presented in Equation (1) should be redefined, respectively, as $\max\{j \le m \mid \sum_{k=i+1}^{j} s_k \le c_u + \Delta c_u(\varepsilon)\}$ and $\max\{v \le n \mid \sum_{k=i+1}^{\texttt{LastOt}(v)} s_k \le c_u + \Delta c_u(\varepsilon)\}$. Moreover, the third case in Equation (1) when $j \neq i$ should be replaced with $E(v,j) + \tilde{e} + \sum_{k=i+1}^{j} s_k \times (e_u + \Delta e_u(\varepsilon))$. Given a specific estimation error ε, the corresponding DFSP$_\varepsilon$ problem can be solved by Algorithm 1 to derive an optimal data fetch schedule σ_ε. Thus, the minimum energy consumption required for data reception based on the fetch schedule σ_ε can be expressed by

$$f(\varepsilon) = \sum_{\substack{\forall p_u \in P \\ \sigma_\varepsilon(u) \neq \{\phi\}}} \left(\tilde{e} + \sum_{\forall o_k \in \sigma_\varepsilon(u)} s_k \times (e_u + \Delta e_u(\varepsilon)) \right),$$

where $\Delta e_u(\varepsilon)$ is the additional energy cost incurred by the estimation error, compared with the objective function defined in Section 2.

Since the energy model and the data rate model are monotonic, nonlinear functions of signal strength, as indicated in Figure 2, performing the postoptimal analysis over a given continuous range of estimation error involves solving an infinite number of DFSP$_\varepsilon$ problems. To make the analysis possible, we assume that, at each checking location p_u, the energy cost e_u and the data rate d_u vary with estimation error at constant rates of Δe_u (joules per byte)/dBm and Δd_u (bytes per second)/dBm, respectively. In other words, Δe_u and Δd_u are the respective slopes of the energy model and the data rate model at the corresponding signal strength. Moreover, the maximum fetch size c_u is proportional to the data rate d_u, so its varying rate, denoted as Δc_u bytes/dBm, is also a constant. Consequently, $\Delta e_u(\varepsilon)$ and $\Delta c_u(\varepsilon)$ can be approximated by linear forms, $\varepsilon \Delta e_u$ and $\varepsilon \Delta c_u$, respectively. Note that the monotonic characteristic (i.e., non-increasing or non-decreasing) of the energy and data rate models should still hold under the linear assumption. That is, for any two checking locations p_u and p_v, if $e_u \le e_v$, then $e_u + \varepsilon \Delta e_u \le e_v + \varepsilon \Delta e_v$; similarly, if $d_u \ge d_v$, then $d_u + \varepsilon \Delta d_u \ge d_v + \varepsilon \Delta d_v$. To ensure the monotonic characteristic, it is apparent that the postoptimal analysis over ε is analyzable only in a range $[\underline{\varepsilon}, \overline{\varepsilon}]$ dBm[3], where the lower bound

$$\underline{\varepsilon} = \max_{1 \le u,v \le n} \begin{cases} -\frac{e_u - e_v}{\Delta e_u - \Delta e_v}, & \text{if } \Delta e_u - \Delta e_v < 0, \\ -\frac{d_u - d_v}{\Delta d_u - \Delta d_v}, & \text{if } \Delta d_u - \Delta d_v > 0, \end{cases} \quad (2)$$

and the upper bound

$$\overline{\varepsilon} = \min_{1 \le u,v \le n} \begin{cases} -\frac{e_u - e_v}{\Delta e_u - \Delta e_v}, & \text{if } \Delta e_u - \Delta e_v > 0, \\ -\frac{d_u - d_v}{\Delta d_u - \Delta d_v}, & \text{if } \Delta d_u - \Delta d_v < 0. \end{cases} \quad (3)$$

With the linear assumption, we will prove in Section 4.3 that the function $f(\varepsilon)$ over the range $[\underline{\varepsilon}, \overline{\varepsilon}]$ is a piecewise linear function. As a result, we need only to solve a finite number of DFSP$_\varepsilon$ problems, for some $\varepsilon \in [\underline{\varepsilon}, \overline{\varepsilon}]$. Let ε_h, for $1 \le h \le r$, denote those *critical points* at which the optimal fetch schedule derived by Algorithm 1 changes. That is, for any two adjacent critical points, we have two important properties: (1) their optimal fetch schedules differ from each other, and (2) function $f(\varepsilon)$ is linear over the range between them. To perform the postoptimal analysis, it is sufficient to solve DFSP$_\varepsilon$ at those critical points as well as the lower and upper bounds, i.e., $\varepsilon = \underline{\varepsilon}, \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_r$, and $\overline{\varepsilon}$.

Given an optimal schedule $\sigma_{\hat{\varepsilon}}$ for DFSP$_{\hat{\varepsilon}}$, we show how to find the critical point immediately smaller than $\hat{\varepsilon}$. In other words, we attempt to find the minimum estimation error $\varepsilon \in [\underline{\varepsilon}, \overline{\varepsilon}]$ such that $\sigma_{\hat{\varepsilon}}$ remains an optimal schedule over the range $[\varepsilon, \hat{\varepsilon}]$. To this end, we introduce the *feasibility condition*, expressed by Equation (5), and the *minimum condition*, expressed by Equation (6). Actually, the feasibility condition is the fetch constraint, which is the only constraint related to signal strength among the three constraints described in Section 2. To explain the minimum condition, we define *reschedulable objects*. For $\sigma_{\hat{\varepsilon}}$, let $\texttt{ResOt}(u,v)$ denote the set of objects that can be rescheduled from p_u to p_v such that the resulting schedule is feasible over $[\varepsilon, \hat{\varepsilon}]$. We delineate two cases, depending on $u < v$ or $u > v$.

1) If $u < v$, to ensure the availability constraint, the objects that can be moved away from p_u start with the last object in $\sigma_{\hat{\varepsilon}}(u)$ until immediately before p_v becomes unreachable from p_u; meanwhile, to ensure the fetch constraint, the objects that can be moved into p_v start with the last object in $\sigma_{\hat{\varepsilon}}(u)$ until immediately before p_v exceeds its maximum fetch size. Moreover, to ensure the dispatch constraint, $\texttt{ResOt}(u,v)$ is an empty set if there exists any fetching locations between p_u and p_v in $\sigma_{\hat{\varepsilon}}$, and represents the intersection of the above two sets otherwise.
2) If $u > v$, the objects that can be moved away from p_u are all the objects in $\sigma_{\hat{\varepsilon}}(u)$; meanwhile, to ensure the fetch constraint, the objects that can be moved into p_v start with the first object in $\sigma_{\hat{\varepsilon}}(u)$ until immediately before p_v exceeds its maximum fetch size. Moreover, to ensure the dispatch constraint, $\texttt{ResOt}(u,v)$ is an empty set if there exists any fetching locations between p_v and p_u in $\sigma_{\hat{\varepsilon}}$, and represents the intersection of the above two sets otherwise.

The minimum condition states that, for any two checking locations p_u and p_v in $\sigma_{\hat{\varepsilon}}$, $1 \le u \neq v \le n$, moving $\texttt{ResOt}(u,v)$ from p_u to p_v must not produce a feasible schedule with less energy consumption. Specifically, if p_v is not a fetching location in $\sigma_{\hat{\varepsilon}}$ and the objects fetched at p_u cannot be rescheduled to p_v altogether, then the difference in energy consumption between p_u and p_v for fetching $\texttt{ResOt}(u,v)$ must be less than or equal to a tail energy cost; and if p_v is a fetching location and the objects fetched at p_u can be rescheduled to p_v altogether, then the difference must be less than or equal to a negative tail energy cost. In Section 4.3, we will prove that $\sigma_{\hat{\varepsilon}}$ remains feasible over $[\varepsilon, \hat{\varepsilon}]$ if and only if the feasibility condition holds, and that $\sigma_{\hat{\varepsilon}}$ remains minimum, compared to any feasible schedule over $[\varepsilon, \hat{\varepsilon}]$, if and only if the *minimum condition* holds. In other words, $\sigma_{\hat{\varepsilon}}$ remains optimal over $[\varepsilon, \hat{\varepsilon}]$ if and only if both the conditions hold. Hence, the critical point ε can be derived by solving the linear programming problem defined in Equation (4).

We now present how to find out all the critical points.

3. Fortunately, the signal strength fluctuates at a location are usually within the range $[\underline{\varepsilon}, \overline{\varepsilon}]$, so the postoptimal analysis is applicable to most real-world cases. More details will be discussed in Section 6.

$$\text{Minimize} \quad \varepsilon \in [\underline{\varepsilon}, \overline{\varepsilon}] \tag{4}$$

$$\text{Subject to}$$

$$\sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \leq c_u + \varepsilon \Delta c_u, \ \forall\ 1 \leq u \leq n, \tag{5}$$

$$\sum_{\forall o_k \in \texttt{ResOt}(u,v)} s_k \times (e_u + \varepsilon \Delta e_u - e_v - \varepsilon \Delta e_v) \leq \begin{cases} \tilde{e}, \text{ if } \sigma_{\hat{\varepsilon}}(v) = \{\phi\} \text{ and } \sigma_{\hat{\varepsilon}(u)} \neq \texttt{ResOt}(u,v), \\ -\tilde{e}, \text{ if } \sigma_{\hat{\varepsilon}}(v) \neq \{\phi\} \text{ and } \sigma_{\hat{\varepsilon}(u)} = \texttt{ResOt}(u,v), \end{cases} \quad \forall\ 1 \leq u \neq v \leq n. \tag{6}$$

First, we solve $\text{DFSP}_{\overline{\varepsilon}}$ by Algorithm 1 to obtain $\sigma_{\overline{\varepsilon}}$. Next, by substituting $\sigma_{\overline{\varepsilon}}$ for $\sigma_{\hat{\varepsilon}}$ in Equation (4), the critical point ε_r is derived. Then, we start with $\varepsilon_r - \theta$, where θ is an extremely small positive number (i.e., $\theta \to 0^+$), and solve $\text{DFSP}_{\varepsilon_r - \theta}$ to obtain $\sigma_{\varepsilon_r - \theta}$. Eventually, all the critical points can be derived by repeating the above process until $\underline{\varepsilon}$ is reached or there is no feasible solution to $\text{DFSP}_{\varepsilon_1 - \theta}$. Based on the derived critical points, $f(\varepsilon)$ can be expressed as

$$f(\varepsilon) = \sum_{\substack{\forall p_u \in P \\ \sigma_{\epsilon(\varepsilon)}(u) \neq \{\phi\}}} \left(\tilde{e} + \sum_{\forall o_k \in \sigma_{\epsilon(\varepsilon)}(u)} s_k \times (e_u + \varepsilon \Delta e_u) \right), \tag{7}$$

where

$$\epsilon(\varepsilon) = \begin{cases} \underline{\varepsilon}, & \text{if } \varepsilon \in [\underline{\varepsilon}, \varepsilon_1), \\ \varepsilon_h, & \text{if } \varepsilon \in [\varepsilon_h, \varepsilon_{h+1}), \text{ for } 1 \leq h < r, \\ \varepsilon_r, & \text{if } \varepsilon \in [\varepsilon_r, \overline{\varepsilon}]. \end{cases}$$

Note that if the estimation error ε lies in the range $[\varepsilon_h, \varepsilon_{h+1})$, where ε_h and ε_{h+1} are two adjacent critical points such that $\varepsilon_h \leq 0 < \varepsilon_{h+1}$, then the schedule derived by Algorithm 1 based on the estimated signal strength Γ is an optimal schedule with respect to the real signal strength Γ_ε.

4.2 A Simple Example

We use the same example in Section 3.2 to better explain the postoptimal analysis. In addition to the original settings, suppose that the respective varying rates of the energy costs $\Delta E = \{-2.8 \times 10^{-8}, -2 \times 10^{-8}, -3.5 \times 10^{-8}, -4 \times 10^{-8}\}$ (joules per byte)/dBm, the varying rates of the data rates $\Delta D = \{1.2, 0.15, 1.3, 1.5\}$ (Kbytes per second)/dBm, and the varying rates of the maximum fetch sizes $\Delta C = \{7.2, 1.5, 9.1, 10.5\}$ Kbytes/dBm.

To begin with, we determine the analyzable range $[-5.926, 10]$ dBm, where the lower bound $\underline{\varepsilon} = -5.926$ is computed based on Equation (2)

$$\underline{\varepsilon} = \max \left(\begin{array}{l} -\frac{(4.9-5.7)\times 10^2}{-2.8-(-2)}, -\frac{(5-5.7)\times 10^2}{-3.5-(-2)}, -\frac{(5.1-5.7)\times 10^2}{-4-(-2)}, \\ -\frac{30-15}{1.2-0.15}, -\frac{25-15}{1.3-0.15}, -\frac{23-15}{1.5-0.15} \end{array} \right)$$

and the upper bound $\overline{\varepsilon} = 10$ is computed based on Equation (3)

$$\overline{\varepsilon} = \min \left(\begin{array}{l} -\frac{(4.9-5)\times 10^2}{-2.8-(-3.5)}, -\frac{(4.9-5.1)\times 10^2}{-2.8-(-4)}, -\frac{(5-5.1)\times 10^2}{-3.5-(-4)}, \\ -\frac{30-25}{1.2-1.3}, -\frac{30-23}{1.2-1.5}, -\frac{25-23}{1.3-1.5} \end{array} \right).$$

Next, the DFSP_{10} problem is solved by Algorithm 1 to obtain $\sigma_{10} = \{\{o_1, o_2, o_3, o_4\}, \{o_5, o_6, o_7\}, \varnothing, \varnothing\}$. In schedule σ_{10}, objects are fetched only at p_1 and p_2. Thus, σ_{10} will remain optimal only if the feasibility condition holds at p_1 and p_2. Based on Equation (5), ε is subject to the first two inequalities in the linear programming problem below. Moreover, to ensure that σ_{10} will remain optimal, the minimum condition must also hold at p_1 and p_2. When $u = 1$, no inequality needs to be satisfied since $\texttt{ResOt}(1,2) = \texttt{ResOt}(1,3) = \texttt{ResOt}(1,4) = \{\phi\}$. When $u = 2$, $\texttt{ResOt}(2,1) = \{\phi\}$, $\texttt{ResOt}(2,3) = \{o_6, o_7\}$, and $\texttt{ResOt}(2,4) = \{\phi\}$, so only $v = 3$ needs to be considered. Since $\sigma_{10}(3) = \{\phi\}$ and $\sigma_{10}(2) \neq \texttt{ResOt}(2,3)$, according to Equation (6), ε is subject to the third inequality in the problem below. Consequently, the minimum estimation error ε such that σ_{10} remains optimal over the range $[\varepsilon, 10]$ can be found by solving the following problem:

$$\begin{array}{l} \text{Minimize} \quad \varepsilon \in [-5.926, 10], \\ \text{Subject to} \\ 45 + 60 + 50 + 40 \leq 180 + \varepsilon \times 7.2, \\ 70 + 35 + 50 \leq 150 + \varepsilon \times 1.5, \\ (5.7 - 5) \times 10^2 + \varepsilon \times (-2 + 3.5) \leq \dfrac{0.5 \times 10^8}{(35 + 50) \times 2^{10}}. \end{array}$$

The derived $\varepsilon = 3.333$ dBm is a critical point. Over the range $[3.333, 10]$, Algorithm 1 will always output σ_{10} and, based on Equation (7), function $f(\varepsilon)$ can be expressed by

$$\begin{aligned} f(\varepsilon) &= 0.5 + 195 \times 2^{10} \times (4.9 \times 10^{-6} - \varepsilon \times 2.8 \times 10^{-8}) \\ &+ 0.5 + 155 \times 2^{10} \times (5.7 \times 10^{-6} - \varepsilon \times 2 \times 10^{-8}) \\ &= 2.883 - 0.009\varepsilon. \end{aligned}$$

Then, repeating the above process, we obtain that $\sigma_{3.333-\theta} = \{\{o_1, o_2, o_3, o_4\}, \{o_5\}, \{o_6, o_7\}, \varnothing\}$ remains optimal over the range $\varepsilon \in [2.083, 3.333)$ and $f(\varepsilon) = 3.322 - 0.01\varepsilon$, and $\sigma_{2.083-\theta} = \{\{o_1, o_2, o_3\}, \{o_4, o_5\}, \{o_6, o_7\}, \varnothing\}$ remains optimal over the range $\varepsilon \in [-3.472, 2.083)$ and $f(\varepsilon) = 3.355 - 0.01\varepsilon$. The analysis ends because there is no feasible solution to $\text{DFSP}_{-3.472-\theta}$; it implies that there is no value for $f(\varepsilon)$ over the range $[-5.926, -3.472)$. Finally, $f(\varepsilon)$ over the analyzable range $[-5.926, 10]$ can be expressed by

$$f(\varepsilon) = \begin{cases} 3.355 - 0.01\varepsilon, & \text{if } \varepsilon \in [-3.472, 2.083), \\ 3.322 - 0.01\varepsilon, & \text{if } \varepsilon \in [2.083, 3.333), \\ 2.883 - 0.009\varepsilon, & \text{if } \varepsilon \in [3.333, 10]. \end{cases}$$

Figure 5 illustrates the difference in energy consumption between the schedules derived based on the real signal strength and the estimated signal strength when the former is ε dBm stronger than the latter, i.e., $f(\varepsilon) - f(0)$, for $\varepsilon = [-5.926, 10]$. When the real signal strength is stronger than the estimated signal strength, i.e., $\varepsilon > 0$, the real energy consumption is less than estimated (which implies a negative difference). However, if $\varepsilon < 0$, the real energy consumption is more than the amount estimated. As ε changes in the

range $[-5.926, 10]$, the optimal fetch schedule changes at $\varepsilon = 2.083$ and $\varepsilon = 3.333$. For example, when the real signal strength is 2.5 dBm stronger than the estimated strength, the difference between the actual energy consumption $f(2.5)$ and the estimated energy consumption $f(0)$ is $f(2.5) - f(0) = 3.322 - 0.01 \times 2.5 - 3.355 = -0.058$ joules. Note that when ε is within the range $[-3.472, 2.083)$, it does not matter whether Algorithm 1 is applied to the real signal strength or the estimated signal strength because the derived schedules are the same.

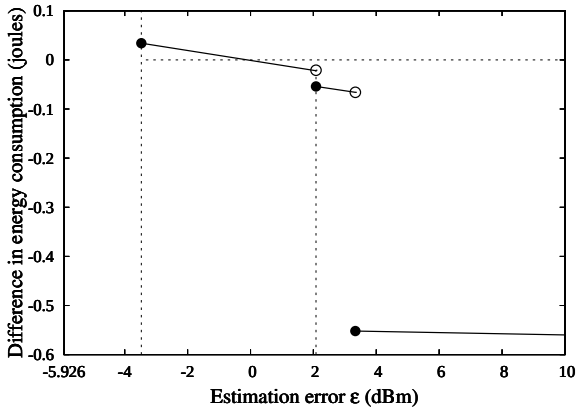


Fig. 5. The difference in energy consumption when the real signal strength is ε dBm stronger than the estimated signal strength

4.3 Properties

***Lemma** 2: The fetch schedule $\sigma_{\hat{\varepsilon}}$ derived by Algorithm 1 for DFSP$_{\hat{\varepsilon}}$ remains feasible over range $[\varepsilon, \hat{\varepsilon}]$ if and only if the feasibility condition expressed by Equation* (5) *holds.*

Proof: Since $\sigma_{\hat{\varepsilon}}$ is a feasible schedule for DFSP$_{\hat{\varepsilon}}$, it must satisfy the dispatch, availability, and fetch constraints defined in Section 2. If $\sigma_{\hat{\varepsilon}}$ remains feasible over $[\varepsilon, \hat{\varepsilon}]$, then the accumulated sizes of the scheduled objects at all checking locations must be less than the corresponding maximum fetch sizes, i.e., $\sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \leq c_u + \varepsilon \Delta c_u, \forall\ 1 \leq u \leq n$.

On the other hand, once $\sigma_{\hat{\varepsilon}}$ becomes infeasible under estimation error ε, there must exist a checking location p_u where the fetch constraint is violated, because only the fetch constraint is related to signal strength. Specifically, $\exists\ 1 \leq u \leq n$ such that $\sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k > c_u + \varepsilon \Delta c_u$. Hence, the proof is completed. □

***Lemma** 3: The feasible fetch schedule $\sigma_{\hat{\varepsilon}}$ derived by Algorithm 1 for DFSP$_{\hat{\varepsilon}}$ remains minimum, compared to any feasible schedule over range $[\varepsilon, \hat{\varepsilon}]$, if and only if the minimum condition expressed by Equation* (6) *holds.*

Proof: If $\sigma_{\hat{\varepsilon}}$ remains minimum compared to any feasible schedule over $[\varepsilon, \hat{\varepsilon}]$, moving $\texttt{ResOt}(u, v)$ from a checking location p_u to another p_v, $1 \leq u \neq v \leq n$, must not produce a feasible schedule with less energy consumption. If p_v is not a fetching location in $\sigma_{\hat{\varepsilon}}$ and the objects fetched at p_u cannot be rescheduled to p_v altogether, then the moving will incur a tail energy cost at p_v. Thus, if $\sigma_{\hat{\varepsilon}}$ remains minimum, the difference in energy consumption between p_u and p_v for fetching the reschedulable objects must be less than or equal to the tail energy cost; otherwise, the moving will produce a feasible schedule with less energy consumption. Specifically, $\forall\ 1 \leq u \neq v \leq n$, if $\sigma_{\hat{\varepsilon}}(v) = \{\phi\}$ and $\sigma_{\hat{\varepsilon}}(u) \neq \texttt{ResOt}(u, v)$ then

$$\sum_{\forall o_k \in \texttt{ResOt}(u,v)} s_k \times (e_u + \varepsilon \Delta e_u - e_v - \varepsilon \Delta e_v) \leq \tilde{e}.$$

If p_v is a fetching location in $\sigma_{\hat{\varepsilon}}$ and the objects fetched at p_u can be rescheduled to p_v altogether, then the moving will have the original tail energy cost at p_u saved. With a similar argument, we have, $\forall\ 1 \leq u \neq v \leq n$, if $\sigma_{\hat{\varepsilon}}(v) \neq \{\phi\}$ and $\sigma_{\hat{\varepsilon}}(u) = \texttt{ResOt}(u, v)$ then

$$\sum_{\forall o_k \in \texttt{ResOt}(u,v)} s_k \times (e_u + \varepsilon \Delta e_u - e_v - \varepsilon \Delta e_v) \leq -\tilde{e}.$$

Hence, if $\sigma_{\hat{\varepsilon}}$ remains minimum over $[\varepsilon, \hat{\varepsilon}]$, then Equation (6) holds.

On the other hand, we shall prove that if Equation (6) holds, then $\sigma_{\hat{\varepsilon}}$ remains minimum over $[\varepsilon, \hat{\varepsilon}]$. Suppose for contradiction that $\sigma_{\hat{\varepsilon}}$ is not minimum under ε. Consequentially, there must exist a pair of p_u and p_v in $\sigma_{\hat{\varepsilon}}$, $1 \leq u \neq v \leq n$, such that moving some objects from p_u to p_v will produce a feasible schedule with less energy consumption. We delineate two cases, depending on whether p_v is a fetching location. When $\sigma_{\hat{\varepsilon}}(v) = \{\phi\}$, if $\sigma_{\hat{\varepsilon}}(u) = \texttt{ResOt}(u, v)$, then moving $\texttt{ResOt}(u, v)$ will not incur or save any tail energy cost. It implies that $e_u + \hat{\varepsilon} \Delta e_u \leq e_v + \hat{\varepsilon} \Delta e_v$ because $\sigma_{\hat{\varepsilon}}$ is optimal under $\hat{\varepsilon}$ and $\texttt{ResOt}(u, v)$ is not empty. Due to the energy model's monotonic characteristic, we have $e_u + \varepsilon \Delta e_u \leq e_v + \varepsilon \Delta e_v$ holds for $\varepsilon \in [\underline{\varepsilon}, \hat{\varepsilon}]$, indicating that moving any objects from p_u to p_v will never produce a feasible schedule with less energy consumption. Additionally, if $\sigma_{\hat{\varepsilon}}(u) \neq \texttt{ResOt}(u, v)$, moving any subset of $\texttt{ResOt}(u, v)$ from p_u to p_v will never produce a feasible schedule with less energy consumption as well, because Equation (6) holds. A similar argument is also applicable for the case when $\sigma_{\hat{\varepsilon}}(v) \neq \{\phi\}$. Hence, a contradiction is reached, which completes the proof. □

***Lemma** 4: The objective function $f(\varepsilon)$ over the analyzable range $[\underline{\varepsilon}, \overline{\varepsilon}]$ is a piecewise linear function.*

Proof: We prove this lemma by showing that $f(\varepsilon)$ is linear over any given range, where no critical point exists between two ends. Since no critical point exists within the range, the schedule, say $\sigma_{\hat{\varepsilon}}$, derived by Algorithm 1 will remain optimal over the range. Then, $f(\varepsilon)$ over the range can be expressed by

$$\sum_{\substack{\forall p_u \in P \\ \sigma_{\hat{\varepsilon}}(u) \neq \{\phi\}}} \left(\tilde{e} + \sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \times (e_u + \varepsilon \Delta e_u) \right),$$

where all of the variables, except ε, are invariant over the range. Hence, let $g(\varepsilon)$ denote

$$\sum_{\substack{\forall p_u \in P \\ \sigma_{\hat{\varepsilon}}(u) \neq \{\phi\}}} \sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \times \varepsilon \Delta e_u.$$

If $g(\varepsilon)$ is a linear function, then $f(\varepsilon)$ is also a linear function since $f(\varepsilon)$ is equal to $g(\varepsilon)$ plus a constant. A function is linear if it satisfies the linearity property [9], i.e., $g(\alpha x+y) = \alpha g(x)+ g(y)$. We obtain that

$$\begin{aligned} g(\alpha x+y) &= \sum_{\substack{\forall p_u \in P \\ \sigma_{\hat{\varepsilon}}(u)\neq\{\phi\}}} \sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \times (\alpha x+y)\Delta e_u \\ &= \alpha \sum_{\substack{\forall p_u \in P \\ \sigma_{\hat{\varepsilon}}(u)\neq\{\phi\}}} \sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \times x\Delta e_u \\ &+ \sum_{\substack{\forall p_u \in P \\ \sigma_{\hat{\varepsilon}}(u)\neq\{\phi\}}} \sum_{\forall o_k \in \sigma_{\hat{\varepsilon}}(u)} s_k \times y\Delta e_u \\ &= \alpha g(x) + g(y). \end{aligned}$$

Thus, $g(\varepsilon)$ is a linear function, which completes the proof. □

***Theorem** 2: The minimum energy consumption $f(\varepsilon)$ for any estimation error ε within the analyzable range $[\underline{\varepsilon}, \overline{\varepsilon}]$ can be expressed by Equation* (7)*.*

Proof: This theorem follows directly from Theorem 1 and Lemmas 2, 3, and 4. □

5 Implementation Remarks

In this section, we discuss some implementation issues that arise when introducing signal strength into location-based applications for energy savings. We have developed a virtual tour system that integrates Google Maps, Panoramio, and YouTube to provide tour information. In addition, it maintains a database that stores signal strength information. The signal strength can be measured in several ways, depending on the circumstances. In crowded downtown areas, the signal strength could be measured and updated frequently via application programs installed by mobile users, e.g., by OpenSignalMaps. The signal strength information in suburbs could be measured by street cars, similar to the way Google collects street views. For mountainous areas, appropriate propagation models, along with the locations of base stations, could be used to estimate the signal strengths. In our prototype system, we used the street-car approach to measure the signal strength in two small areas of Taipei City for our case studies. Once this concept is embraced gradually by location-based service providers, some entities might have intention to collect and provide such information.

We have also developed an augmented reality application program for Android devices to access the tour guide service. When a user starts his/her journey, the program sends the server a request (approx. 246 bytes) specifying the geographic coordinates of the source and destination, as well as the types of information required. On receipt of the user's request, the server recommends a route and determines a fetch schedule by running the proposed algorithm. Then, the route (which consists of a few hundred bytes) with the locations scheduled to fetch information is returned to the program. When the user arrives at a fetching location, the program sends a notification (approx. 88 bytes) to the server in order to fetch the corresponding information (ranging from dozens of Kbytes to a few Mbytes). The user can then browse the information on the device screen. With such a design, mobile devices are exempted from additional computational overheads, and the time required to determine a fetch schedule only ranges from dozens to hundreds of milliseconds on the server. Obviously, all the input parameters required by the algorithm can easily be acquired, except the energy costs and the maximum fetch sizes that are related to the signal strength. In the remainder of this section, we focus on how we estimated the energy costs and determined the maximum fetch sizes at checking locations.

5.1 Energy Cost Estimation

The energy cost (joules per byte) at a checking location is defined as the mobile device's power consumption (watts) divided by the downlink data rate (bytes per second). The downlink data rate has a strong relationship with the signal strength. To plot their relationship, we installed the application program developed by OpenSignalMaps on an HTC EVO 3D smartphone to measure the signal strengths and data rates at various locations in Taipei City. We gathered over 3000 pairs of such data within the coverage of 3G/3.5G signals provided by Chunghwa Telecom. Then, we applied the *polynomial regression method* [24] to the gathered data and modeled the relationship with a monotonic function, as shown in Figure 2. It is no doubt that the more (and diverse) the data gathered, the more accurate the monotonic function, and the less the effect due to signal fluctuations. Furthermore, we observed that the signal strength at a location is generally stable over time, which also agrees with the phenomenon observed in [21]. Based on our measurement, the signal strength at a checking location is close to the expected signal strength with a standard deviation up to 4 dBm, and the standard deviations are smaller than 2 dBm at most locations.

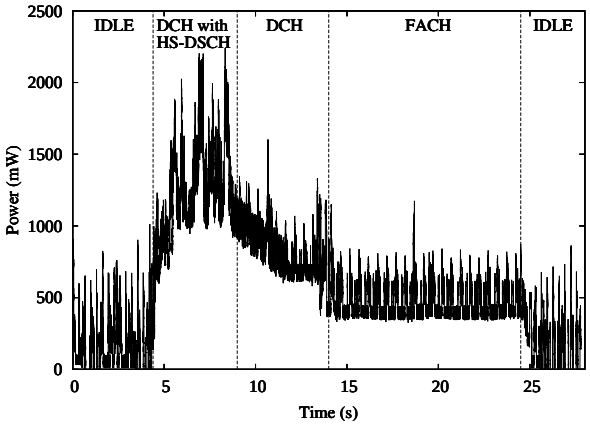


Fig. 6. The power consumption along with the state transition of 3G/3.5G

The power consumption depends mainly on the communication chip adopted by the mobile device. Fortunately, the accuracy of the power model will only affect the amount of energy saved if the scheduled objects can be fetched successfully at every checking location; therefore, other device models could also benefit even if their accurate power models have not be acquired. The receive mode of 3G/3.5G has

four/five states, and the state transition adheres to the *radio resource control protocol* specified in UMTS/HSPA of the 3GPP standard. In practice, we used the power monitor produced by Monsoon Solutions to measure the power consumption of the HTC EVO 3D smartphone. Figure 6 shows the power consumption of each state during an ICMP ping. The radio interface is initiated in CELL_IDEL, which consumes almost no power. Then, it transits to CELL_DCH with HS-DSCH, a state supporting high-speed data downlink, and consumes 1050 mW when remaining in the state for data reception. Thus, the energy cost at a location can be computed by dividing 1.05W by the downlink data rate there. After that, the interface starts to release the radio resources, resulting in a state demotion, and lasts in CELL_DCH with power consumption of 590 mW until an inactive timer of 5 seconds expires. Again, the radio interface remains in CELL_FACH until another expiration of 12 seconds, and eventually returns to the idle state. The power consumption in CELL_FACH is 310 mW. Thus, the tail energy cost can be computed by $0.59 \times 5 + 0.31 \times 12 = 6.67$ (joules).

5.2 Maximum Fetch Size Determination

The maximum fetch size (bytes) at a checking location is defined as the downlink data rate (bytes per second) multiplied by the time (second) the mobile device stays in the *effective region* covered by the same signal strength around that location. The effective region depends on the distance between the mobile device and the base station, surrounding obstacles, interference from other cells, among others [10]. To capture the changes of the signal strengths along a route, we installed a mobile application program, called RF Signal Tracker, to measure the signal strengths and their effective regions around the checking locations along the route. For example, the red circles in Figure 7(a) indicate the effective regions of the corresponding checking locations. Note that the effective radiuses may be different from location to location, varying from 19 meters to 75 meters, depending on the signal strengths and surrounding circumstances. Based on our measurement, the effective radiuses centered at checking locations along a route are 45 meters on average.

(a) Effective regions (b) The mobile platform

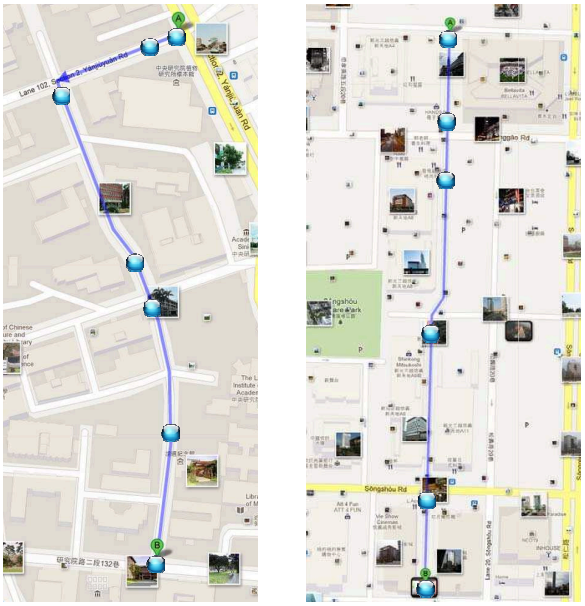
Fig. 7. Experiment equipments

The time a mobile device stays in an effective region depends on the user's movement speed. Based on some statistics, the velocities of pedestrians [13], bicyclers [7], and drivers [22] in urban areas are roughly 83, 216, and 667 meters per minute, respectively. In other words, the respective users would take roughly 65, 25, and 8.1 seconds to pass through an effective region with a radius of 45 meters for example. The maximum fetch sizes can be computed accordingly. To ensure that the scheduled objects can always be fetched successfully before the user leaves the effective region, we would prefer underestimating the maximum fetch sizes slightly. We simulate different velocities by varying the maximum fetch sizes and discuss the impact on energy savings in Section 6.

6 Performance Evaluation

6.1 Experiment Setup

To demonstrate the efficacy of introducing signal strength into location-based applications, we conducted a series of experiments in Taipei City. In the experiment platform, shown in Figure 7, the mobile application program was installed on an Android smartphone of HTC EVO 3D, equipped with a 3.5G communication subsystem, and the energy consumption was measured by the Power Monitor of Monsoon Solutions. We evaluated the performance of the proposed algorithm, denoted as OPT, in terms of the energy consumption required for data reception. To show the performance gain, we compared OPT with the native approach, denoted as NATIVE, adopted by Google Maps. We also estimated the energy consumption based on our system model, denoted as OPT-THEORY, to gain further insights into the gap between theoretical estimation and experiment measurement.



(a) Route@campus (b) Route@downtown

Fig. 8. Two real-world case studies in Taipei City

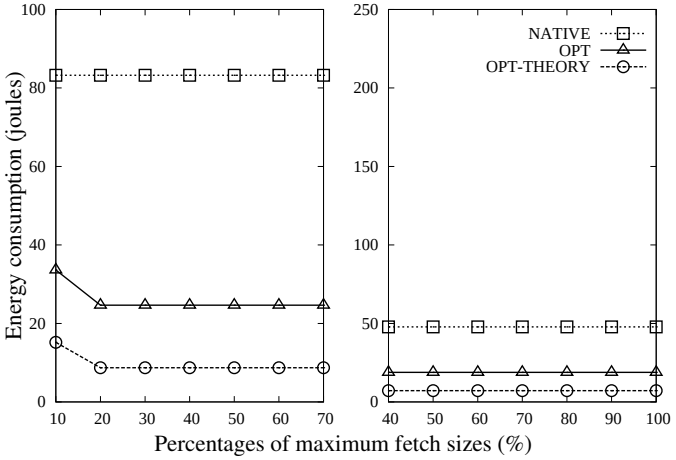
We investigated two routes with diverse characteristics, as shown in Figure 8. Route@campus is a path in Academia Sinica located in the suburb of Taipei City, while Route@downtown represents a crowded street in an urban

area like the Xinyi District. Route@campus has seven checking locations and Route@downtown has six. The location-based information is relatively sparse along Route@campus (i.e., 54 objects, including 24 maps tiles, 7 street views, 22 photos, and 1 video); while it is dense along Route@downtown (i.e., 239 objects, including 21 map tiles, 1 street view, 214 photos, and 3 videos). The respective sizes of a Google map tile (/street view), Panoramio photo, and YouTube video are approximately 10 (/30), 30, and 4000 Kbytes. The additional bytes, presented in the form of {Google map tiles and street views, Panoramoio photos, YouTube videos}, needed for each checking location are {106, 130, 0}, {63, 122, 0}, {51, 27, 0}, {57, 144, 4006}, {48, 132, 0}, {54, 94, 0}, and {38, 133, 0} along Route@campus, and {103, 1312, 0}, {13, 1304, 0}, {17, 115, 2265}, {14, 817, 2485}, {12, 1094, 0}, and {0, 879, 4121} along Route@downtown. On the other hand, the signal is relatively weak at the checking locations along Route@campus (i.e., -77, -75, -78, -86, -79, -91, and -91 dBm, with -82 dBm on average) and strong along Route@downtown (i.e., -65, -72, -78, -76, -58, and -60 dBm, with -68 dBm on average). The corresponding data rate at each checking location was derived based on the model in Figure 2, and the maximum fetch size was computed based on the corresponding effective region and the velocity of pedestrians (i.e., 83 meters per minute).

To explore the impact of the amount of location-based information, we considered three scenarios. LBS3 provided all the three applications, namely Google Maps, Panoramio, and YouTube; LBS2 provided the first two and LBS1 provided only the first one. For each scenario, we were supposed to consider various mobile users (e.g., pedestrians, bicyclers, and drivers). However, we could not wheel the experiment platform at various velocities. Thus, we simulated a variety of velocities by underestimating the maximum fetch sizes, and measured the energy consumption of the fetch schedules derived by Algorithm 1 towards different percentages of underestimation. Moreover, we carried out postoptimal analysis to explore the tolerance of the proposed algorithm to signal strength fluctuations under each scenario. Derived by Equations (2) and (3), the analyzable ranges for Route@campus and Route@downtown are $[-5.048, 16.703]$ dBm and $[-14.21, 7.689]$ dBm, both of which are larger than the range of potential strength fluctuations (usually up to ± 4 dBm). For ease of comparison, we performed postoptimal analysis over their common analyzable range, namely $[-5.048, 7.689]$ dBm. The analysis delivers some important insights, especially the impact of estimation error ε on the proposed algorithm's optimality and performance.

6.2 Experiment Results

Figures 9(a) and 9(b) show, respectively, the energy consumption required by LBS1 along Route@campus and Route@downtown. The maximum fetch sizes have no impact on the performance of NATIVE, because it always fetches the corresponding objects when a local scene is needed. In contrast, for OPT, it is expected that the energy consumption will decrease as the maximum fetch sizes increase, because larger fetch sizes imply more flexibility. Interestingly, the results are not exactly as expected. It is because LBS1 comprises



(a) Route@campus (b) Route@downtown

Fig. 9. Energy consumption required by LBS1

only a few small objects. Consequently, most objects can be fetched at a few specific locations with stronger signal. This also explains why the energy consumption required by LBS1 is less along Route@downtown than along Route@campus. The results show that, with the information of signal strength, OPT reduces the communication energy by 59-70% along Route@campus and about 61% along Route@downtown, compared to that consumed by LBS1 under NATIVE. We also observed a gap that cannot be ignored between OPT and OPT-THEORY. One reason for the gap is that the measured signal strengths and the energy model could not be one hundred percent accurate. Other reasons could include the overheads of packet headers/control packets, the round-trip time of requests, and the retransmission of data, which are not considered in our system model. Finally, there are feasible schedules even if the maximum fetch sizes are very small. It implies that LBS1 is applicable to mobile users with a variety of velocities.

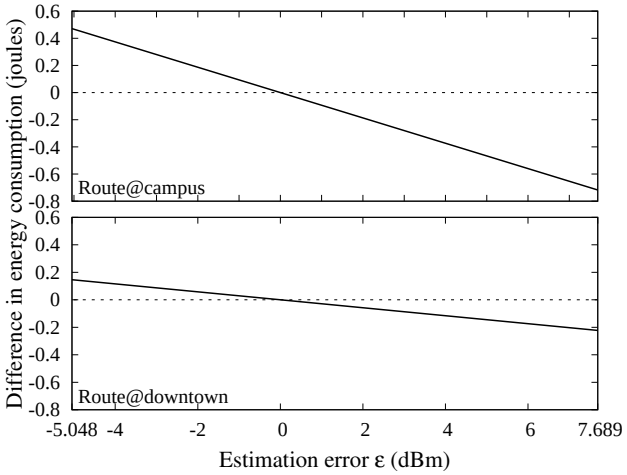
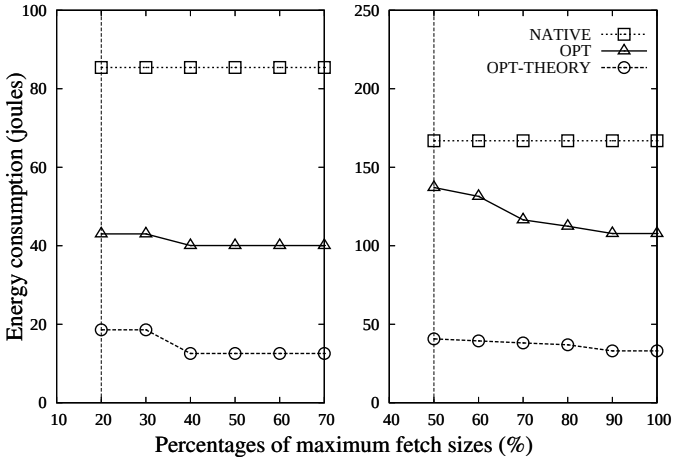


Fig. 10. Postoptimal analysis for LBS1

Figure 10 shows the postoptimal analyses for LBS1 along the two routes, respectively. As expected, the difference in energy consumption decreases as the estimation error increases, because the stronger the real signal strength is, the less the energy will really be consumed. Moreover, the respective decreasing rates of Route@campus and Route@downtown

are -0.093 and -0.029 joules/dBm which are considered very small. Note that the decreasing rate represents the impact of the estimation error on the difference in energy consumption between the real signal strength and the estimated signal strength. Thus, the small decreasing rates tell us that the signal strength fluctuations contribute only slightly to the gaps between OPT and OPT-THEORY in Figures 9(a) and 9(b), while the other factors not considered in our system model contribute more relatively. On the other hand, the analysis indicates that both the fetch schedules derived along the two routes remain optimal over the entire analyzed range $[-5.048, 7.689]$ dBm. In other words, for LBS1 along the two routes, there is no difference no matter the proposed algorithm is conducted on the real signal strength or the estimated signal strength.



(a) Route@campus (b) Route@downtown

Fig. 11. Energy consumption required by LBS2

Figures 11(a) and 11(b) show the respective energy consumption required by LBS2 along the two routes. The energy consumption under OPT decreases as the maximum fetch sizes increase, and the decrease is more manifest for Route@downtown than for Route@campus. This is because, along Route@downtown, LBS2 comprises many objects and the signal strengths vary more significantly. When the maximum fetch sizes are larger, OPT has more flexibility to schedule objects based on signal strength, resulting in more energy savings. In addition, there are much more objects along Route@downtown than along Route@campus; thus, the energy consumption is more along Route@downtown even if the signal is stronger. We observed a large discrepancy between OPT and OPT-THEORY. The main reason is that LBS2 contains many photos, and fetching a photo requires a round-trip time during which the radio interface remains in the high power state. The discrepancy could be reduced by combining the photos of a local scene into a large object from the perspective of system implementation, or by considering the round-trip time in the system model although the problem definition will become more complex. Despite the disregard of the round-trip time, OPT still achieves great energy savings. The results show that the communication energy is reduced by 49-53% along Route@campus and 18-35% along Route@downtown, depending on the maximum fetch sizes. Finally, as can be seen in the figures, no feasible schedule exists when the maximum fetch sizes are set at small. Thus, LBS2 might be more applicable to pedestrians and bicyclers than drivers.

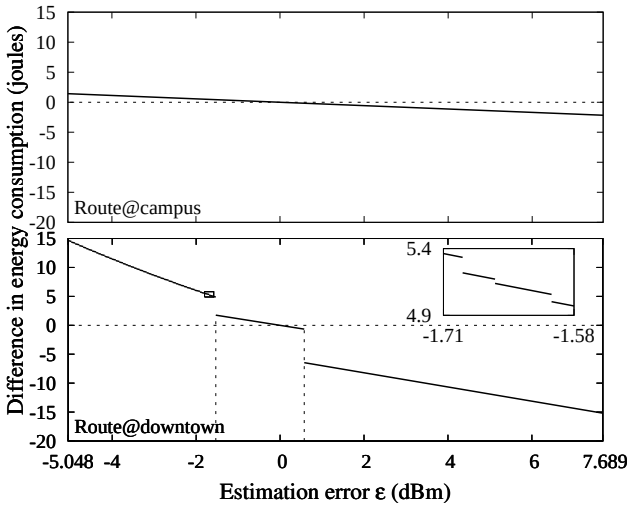
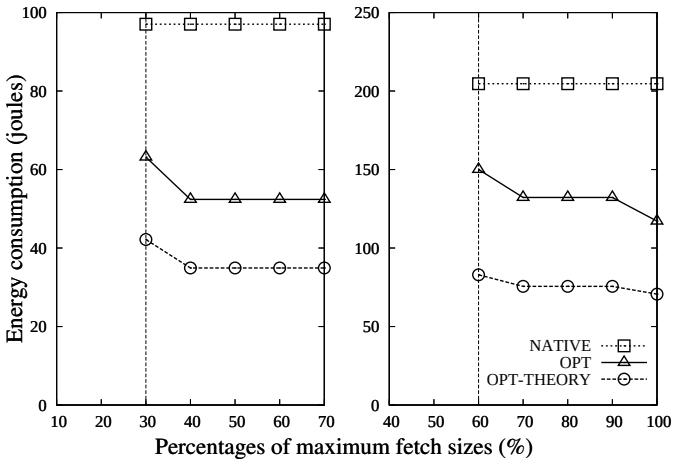


Fig. 12. Postoptimal analysis for LBS2

Figure 12 shows the respective postoptimal analyses for LBS2 along the two route. The decreasing rate of the difference in energy consumption for Route@campus is -0.281 joules/dBm, which is also small and implies that the signal strength fluctuation may not be a decisive factor to the large discrepancy between OPT and OPT-THEORY in Figure 11(a). In contrast, the decreasing rate of Route@downtown grows considerably to -2.314 joules/dBm, compared with the counterpart under LBS1 in Figure 10. The growth is mainly attributed to the hundreds of objects required for LBS2 along Route@downtown. Note that when the object size is large, a slight change in the energy cost will result in a significant change in the energy consumption. Therefore, the discrepancy due to the estimation error occupies a noticeable portion, roughly one-fifth at most, of the gap between OPT and OPT-THEORY in Figure 11(b). Despite the noticeable portion, other factors, especially the round-trip time, still dominate the gap. On the other hand, the fetch schedule derived along Route@campus remains optimal over the entire analyzed range $[-5.048, 7.689]$ dBm. However, the fetch schedule derived along Route@downtown is optimal only within the range $[-1.53, 0.579)$ dBm. Although the derived schedule is not optimal outside the range, it remains a feasible schedule when $\varepsilon \geq 0.579$, because the scheduled objects can still be fetched successfully when the signal becomes stronger (as ε increases). In other words, only the amount of energy saved will be affected, not the application's semantics. For example, when the schedule derived based on the estimated signal strength is applied to the case when $\varepsilon = 4$ dBm, the resulting energy consumption is 6.077 joules more than the optimal solution. However, when $\varepsilon < -1.53$ dBm, the derived fetch schedule becomes infeasible. To address this issue, we can underestimate the maximum fetch sizes (or the estimated signal strength) at the cost of the amount of energy saved. To ensure that the scheduled objects can always be fetched successfully, the maximum fetch sizes should be set as 87% of the original sizes if $\varepsilon \geq -2$ dBm and 75% if $\varepsilon \geq -4$

dBm.



(a) Route@campus (b) Route@downtown

Fig. 13. Energy consumption required by LBS3

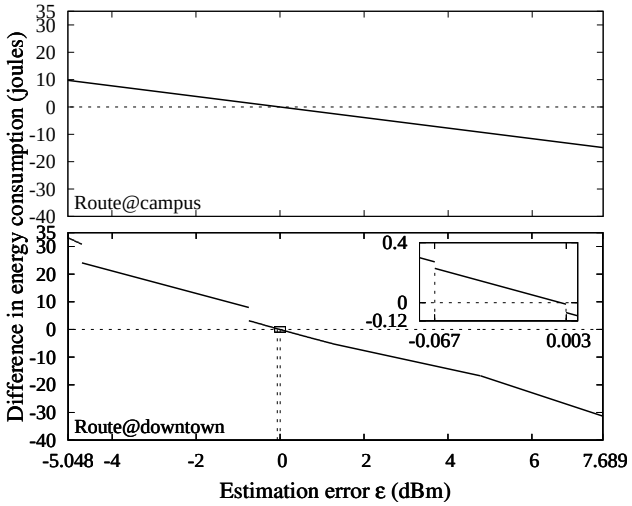


Fig. 14. Postoptimal analysis for LBS3

Figures 13(a) and 13(b) show the energy consumption required by LBS3 along the two routes. Clearly, the videos cause a significant increase in the energy consumption, compared with the energy consumption shown in Figures 9 and 11. However, the increase is not proportional to the increase in the amount of information in LBS3. The reason is that LBS3 contains some large-sized videos, in addition to maps and photos. The additional energy consumption incurred by the round-trip time accounts for a larger proportion of the energy consumed by a small object than by a large object. In addition, more energy savings can be achieved by OPT if the videos can be scheduled to some locations more appropriate than where they are fetched under NATIVE. The results show that the communication energy is reduced by 35-46% along Route@campus and 27-43% along Route@downtown. Moreover, we observed that the gap between OPT and OPT-THEORY is significantly reduced, compared with that in Figure 11. It is because the additional energy consumption incurred by the round-trip time is amortized by the energy consumed by the videos. This also evidences the significant impact of the round-trip time on the communication energy. Finally, there will be no feasible schedule if the maximum fetch sizes are not sufficient to fetch videos. Thus, LBS3 might be applicable to only pedestrians.

Figure 14 shows the respective postoptimal analyses for LBS3 along the two routes. The decreasing rates of the difference in energy consumption for Route@campus and Route@downtown are -1.933 and -5.054 joules/dBm, respectively. The reason for the large decreasing rates is that LBS3 contains some videos of large sizes, as a similar reason explained for LBS2 along Route@downtown. Furthermore, as mentioned above, the additional energy consumption incurred by the round-trip time is amortized by the energy consumed by the videos. With the two phenomena combined, the signal strength fluctuations contribute relatively more to the discrepancy between OPT and OPT-THEORY. For example, when $\varepsilon = -4$ dBm, the additional energy consumption due to the signal strength fluctuations occupies roughly 37% and 42% of the respective gaps in Figures 13(a) and 13(b). On the other hand, like the respective schedules derived for LBS1 and LBS2 along Route@campus, the schedule derived for LBS3 along Route@campus remains optimal over the entire analyzed range $[-5.048, 7.689]$ dBm. Thus, it could be concluded that the proposed algorithm can tolerate signal strength fluctuations very well when the objects along a route are sparse. In contrast, the optimal range of the schedule derived for LBS3 along Route@downtown is narrowed down to $[-0.067, 0.003)$ dBm, compared with the counterpart in Figure 12. It is because the large size videos accelerate the reaching of the maximum fetch sizes at those checking locations with stronger signal, resulting in frequent schedule changes as the fetch sizes are reduced. In fact, the error range within which a derived schedule remains stable depends on the dispatch sequence of the objects and the varying rates of the maximum fetch sizes. As indicated in the figure, the derived schedule remains feasible when $\varepsilon \geq 0.003$ but becomes infeasible when $\varepsilon < -0.067$. To ensure the derived schedule for LBS3 against the signal strength fluctuations along Route@downtown, the maximum fetch sizes should be shrunk at 87% when $\varepsilon \geq -2$ dBm and 75% when $\varepsilon \geq -4$ dBm.

7 Discussions

Through real-world case studies, we have demonstrated the practicability of introducing signal strength into location-based applications. However, much more remains to be done. In this section, we discuss the limitations of our work and highlight issues that require further investigation.

7.1 Checking Location Selection

In our prototype implementation, the checking locations are only situated at the intersections of the route and the boundaries of the map tiles. We use this setting to ensure a fair comparison with the native approach adopted by Google Maps. There may be some locations where the signals are stronger than those at the selected checking locations. If more locations, especially those with good signal strength, are selected as checking locations, it is more likely that a fetch schedule could be found with less energy consumption. However, we cannot select an unlimited number of checking

locations because the proposed algorithm's running time increases with the number of such locations. The algorithm's efficacy and running time could be further improved by selecting appropriate checking locations and eliminating locations that are unnecessary.

7.2 Energy and Data Rate Model Enhancement

We only use the signal strength as the indicator for data rate estimation because our main objective is to understand the impact of the signal strength and its fluctuation on the energy consumption during data reception. Therefore, we only consider a system model that is affected by a single factor in order to investigate the factor's actual impact. In fact, other factors, such as the base station's load and the user's movement speed, have varying degrees of impact on the data rate [8]; hence, they also contribute to the gap between the theoretical estimation and the measurement derived in the experiments, i.e., between OPT-THEORY and OPT. It would therefore be interesting to consider multiple factors and analyze their compound impact on the data rate model.

On the other hand, the energy cost under some signal strengths is estimated by dividing the fixed power consumption of 1050 mW measured in Figure 6 by the corresponding downlink data rate. The actual power consumption under different signal strength may differ slightly from the constant value and it may vary because of certain factors. For example, the power consumption may vary due to *uplink power control* based on signal strength. Because the TCP downlink involves a significant number of uplink ACK packets, the actual power consumption under a weak signal may be more than 1050 mW even though the ACK packets are much smaller than the data packets. The round-trip time also has a significant impact on the downlink data rate and thus the energy cost estimation, as shown by the experiment results. The accuracy of the energy model could be improved by providing a better energy estimate. It would also be interesting to perform postoptimal analysis on errors in the energy cost estimation.

7.3 Dynamic Approaches

Our technical contribution is that we analyze the impact of signal fluctuations on the practicability of applying the proposed concept in real-world environments, rather than develop a fully functional virtual tour system. Accordingly, we exploit a static approach because it offers advantages for analysis. When the concept is applied to existing location-based applications like Google Maps, a dynamic approach should be employed to handle unexpected situations, e.g., when the user abandons part of the route due to missing a turn or significant signal fluctuations are detected. For example, when the user deviates the original route, a dynamic approach could adjust the fetch schedule on-the-fly by applying the proposed algorithm to the new route. Nevertheless, to avoid wasting energy on unnecessary data prefetching, a dynamic approach with sophisticated probability models should be developed to deal with dynamic user behavior. Besides, a dynamic approach is needed to adjust the fetch schedule when the estimation error exceeds the optimal range (derived during postoptimal analysis) due to signal fluctuations.

7.4 Adaptation to LTE

LTE is becoming increasingly popular because it has very high throughput (but the tail energy is still large), so downloading all the information in a single data burst may be possible. However, if the strategy involves downloading everything at a single location, all the data must be downloaded at the source location; otherwise, some local scenes may not be available when the user arrives at the corresponding locations. If the signal strength at the source is weak, energy will be wasted because the communication energy in LTE still depends on the downlink data rate [12] , which is affected by the signal strength [11]. In addition, the emergence of LTE may drive the development of new bandwidth-intensive mobile applications whose information cannot be downloaded in a single data burst. Hence, the data fetch scheduling problem is still relevant in the context of LTE.

8 Related Work

Much of the early research focused on leveraging the complementary characteristics of WiFi and 3G to gain energy efficiency. *Delay-tolerant* applications, like email and blogging, can afford to delay data transfer without significantly hurting user experience. Thus, various techniques have been proposed to trade off transfer latency for energy savings. Prediction-based approaches predict the future availability of WiFi connectivity, and delay data communications via 3G until WiFi becomes available if the delay is tolerable [1, 20]. The rationale behind the energy gains is that, compared to 3G, WiFi consumes much less energy per bit for data communications [19]; thus, WiFi is employed to improve the energy efficiency when available. In contrast, *delay-sensitive* applications, such as video streaming and web mapping, in mobile environments are extremely sensitive to delay and thus rely on ubiquitous wireless connectivity. The approach proposed in [19] selects the more efficient one based on context information. A fast switching mechanism was developed in [2] to enable switching to 3G once WiFi fails. In [18], a theoretic framework for exploring energy-delay tradeoffs was introduced to achieve balance in a wide spectrum between energy efficiency and data delay.

Another research direction aimed at energy-efficient communications in 3G networks. A large fraction of the communication energy in 3G is wasted on the tail energy [3], because 3G does not switch from the high to the low power state immediately after each data transfer; instead, it waits for a period to alleviate the switching overhead, in case another communication will occur before long. The approaches that delay data transfer to reduce cumulative tail energy could be categorized into *batch scheduling* [4, 14] and *fast dormancy* [16, 17]. Recently, it has been observed that signal strength has a direct impact on the communication energy consumption. In [21], a threshold-based approach was proposed for email syncing by deferring communications until a location with better signal strength. For video streaming, a schedule-based approach was presented, in a conceptual manner, to assign

video frames into time slots associated with different signal strength such that the communication energy is minimized. However, the approach relies on an accurate estimate of signal strength, and little attention has been paid to the impact of signal fluctuations on the feasibility of the derived solution.

9 Conclusion

In this paper, we introduce signal strength into location-based applications to reduce the communication energy of mobile devices. The rationale behind the reduction is that the communication energy is much more when the signal is weak than when it is strong. To prove the concept, we have developed a virtual tour system, where the key technology is to schedule appropriate fetching locations for objects based on signal strength. We propose a dynamic-programming algorithm to derive optimal schedules in terms of energy savings, and perform postoptimal analysis to explore the impact of signal strength fluctuations on the proposed algorithm's optimality and performance. To evaluate the improvement in energy efficiency, we conducted a series of experiments along two routes of diverse characteristics in Taipei City. The results show that an HTC EVO 3D smartphone can achieve energy savings of 46-70% and 35-60% for pedestrian users along the two routes, respectively. Moreover, the algorithm can tolerate signal strength fluctuations very well when the objects along a route is sparse. When the impact of signal strength fluctuations is significant, we could slightly underestimate the estimated signal strength (at the cost of the amount of energy saved) to ensure that the scheduled objects can always be fetched successfully.

Acknowledgment

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the National Science Council under grant Nos. 101-2219-E-002-002 and 102-2221-E-001-007-MY2, which the authors gratefully acknowledge.

References

[1] G. Ananthanarayanan and I. Stoica. Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals. In *Proc. of the ACM MobiSys*, pages 249–262, 2009.

[2] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting Mobile 3G using WiFi. In *Proc. of the ACM MobiSys*, pages 209–222, 2010.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Smartphones: A Measurement Study and Implications for Network Applications. In *Proc. of the ACM IMC*, pages 280–293, 2009.

[4] M. Calder and M. K. Marina. Batch Scheduling of Recurrent Applications for Energy Savings on Mobile Phones. In *Proc. of the IEEE SECON*, pages 1–3, 2010.

[5] C.-C. Cheng and P.-C. Hsiu. Extend Your Journey: Introducing Signal Strength into Location-based Applications. In *Proc. of the IEEE INFOCOM*, pages 2742–2750, 2013.

[6] S. Dhar and U. Varshney. Challenges and Business Models for Mobile Location-based Services and Advertising. *Commun. ACM*, 54(5):121–128, 2011.

[7] A. El-Geneidy, K. J. Krizek, and M. Iacono. Predicting Bicycle Travel Speeds along Different Facilities using GPS Data: a Proof of Concept Model. In *Proc. of the TRB*, 2007.

[8] A. Gember, A. Akella, J. Pang, A. Varshavsky, and R. Caceres. Obtaining in-context measurements of cellular network performance. In *Proc. of the ACM IMC*, pages 287–300, 2012.

[9] K. Hoffman and R. A. Kunze. *Linear Algebra*, page 97. Prentice-Hall, second edition, 1961.

[10] H. Holma and A. Toskala. *WCDMA for UMTS: HSPA Evolution and LTE*, pages 175–221. John Wiley & Sons, Inc., fourth edition, 2007.

[11] H. Holma and A. Toskala. *LTE for UMTS: Evolution to LTE Advanced*, pages 257–302. John Wiley & Sons, Inc., second edition, 2011.

[12] J. 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 ACM MobiSys*, pages 225–238, 2012.

[13] R. L. Knoblauch, M. T. Pietrucha, and M. Nitzburg. Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Record*, pages 27–38, 1996.

[14] H. Liu, Y. Zhang, and Y. Zhou. TailTheft: Leveraging the Wasted Time for Saving Energy in Cellular Communications. In *Proc. of the ACM MobiArch*, pages 31–36, 2011.

[15] A. J. Nicholson and B. D. Noble. BreadCrumbs: Forecasting Mobile Connectivity. In *Proc. of the ACM MobiCom*, pages 46–57, 2008.

[16] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing Radio Resource Allocation for 3G Networks. In *Proc. of the ACM IMC*, pages 137–150, 2010.

[17] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. TOP: Tail Optimization Protocol For Cellular Radio Resource Allocation. In *Proc. of the IEEE ICNP*, pages 285–294, 2010.

[18] 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 ACM MobiSys*, pages 255–270, 2010.

[19] A. Rahmati and L. Zhong. Context-Based Network Estimation for Energy-Efficient Ubiquitous Wireless Connectivity. *IEEE Trans. on Mobile Computing*, 10(1):54–66, 2011.

[20] N. Ristanovic, J.-Y. Le Boudec, A. Chaintreau, and V. Erramilli. Energy Efficient Offloading of 3G Networks. In *Proc. of the IEEE MASS*, pages 202–211, 2011.

[21] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: A Practial Approach to Energy-aware Cellular Data Scheduling. In *Proc. of the ACM MobiCom*, pages 85–96, 2010.

[22] A. Studer, M. Luk, and A. Perrig. Efficient Mechanisms to Provide Convoy Member and Vehicle Sequence Authentication in VANETs. In *Proc. of the IEEE SecureComm*, pages 422–432, 2007.

[23] S. J. Vaughan-Nichols. Will Mobile Computing's Future Be Location, Location, Location? *IEEE Computer*, 42(2):14–17, 2009.

[24] S. Weisberg. *Applied Linear Regression*, pages 115–146. Wiley/Interscience, third edition, 2005.

Chih-Chuan Cheng received the BS degree in electrical engineering from Minghsin University of Science and Technology in 2000 and the MS and PhD degrees in electrical engineering from National Chung Cheng University (NCCU) in 2005 and 2011, respectively. In 2010, he was a visiting PhD student in the Department of Electrical and Computer Engineering at San Diego State University. He has been a postdoctoral fellow in the Research Center for Information Technology Innovation, Academia Sinica, Taiwan, since 2011. His research interests include wireless sensor networks and wireless mobile networks.

Pi-Cheng Hsiu received the BS degree in computer information science from National Chiao Tung University in 2002 and the MS and PhD degrees in computer science and information engineering from National Taiwan University in 2004 and 2009, respectively. He is currently an associate research fellow in the Research Center for Information Technology Innovation (CITI), where he leads the Embedded and Mobile Computing Laboratory (EMCLab), and also a joint associate research fellow in the Institute of Information Science (IIS), Academia Sinica, Taiwan. He was a visiting scholar in the Department of Computer Science at the University of Illinois at Urbana-Champaign in 2007 and joined CITI as an assistant research fellow in 2009. His research interests include embedded systems, real-time systems, and mobile networks. He is a member of the IEEE and the ACM.