Effects of Cache Mechanism on Wireless Data Access

Yi-Bing Lin, *Fellow, IEEE*, Wei-Ru Lai, and Jen-Jee Chen

***Abstract*—In wireless data transmission, the capacity of wireless links is typically limited. Since many applications exhibit temporal locality for data access, the cache mechanism can be built in a wireless terminal to effectively reduce the data access time. This paper studies the cache performance of the wireless terminal by considering a business-card application. We investigate the least-recently used replacement policy and two strongly consistent data access algorithms called poll-each-read and callback. An analytic model is proposed to derive the effective hit ratio of data access, which is used to validate against simulation experiments. Our study reports how the data access rate and the data update distribution affect the cache performance in a wireless terminal.**

***Index Terms*—Cache, strong consistency, temporal locality, wireless data.**

I. Introduction

MODERN mobile networks support wireless data applications. In the past, wireless data services were marketed as wireless modem access for laptops or notebooks to offer so called "portable computing." As advanced wireless infrastructures become available and the inexpensive wireless handheld devices (e.g., wireless personal data assistant (PDA) and wireless smart phones) become popular, the mobile subscribers can enjoy instant wireless Internet access. Several wireless application platforms have been developed to support wireless Internet applications. Examples include *wireless application protocol* (WAP), [21], [23], [24] and iSMS [15]. A typical wireless application platform is illustrated in Fig. 1. In this architecture, the wireless application gateway (e.g., the WAP Gateway) interworks the wireless network with the Internet protocol (IP) network, which allows a wireless terminal to obtain data from an application server (e.g., an origin server in WAP). To converge wireless data with the Internet, the WAPs may integrate a light-weight web browser into wireless terminals with limited computing and memory capacities. The WAPs implemented in the wireless application gateway and the wireless terminal enable a mobile user to access Internet web applications through a client–server model. A client application running on the wireless terminal may repeatedly access a data object received from the application server. To speed up wireless data access, cache mechanisms have been proposed. For example, the WAP user agent caching model [22] tailors the hypertext transfer protocol caching model to support wireless terminals with limited functions. In this model, a cache in the wireless terminal is used to buffer frequently used data objects sent from the WAP Gateway. When the data objects in the application server are modified, the cached objects in the wireless terminals are obsolete. To guarantee that the data presented to the user at the terminal are the same as that in the application server, a *strongly consistent* data access protocol [25] must be exercised. Furthermore, the cache size in a wireless terminal is typically small, and a cache replacement policy is required to accommodate appropriate data objects in the cache. We will elaborate on cache replacement and two strongly consistent data access protocols in Section III. Then we utilize both analytic analysis and simulation experiments to investigate how the strongly consistent data access algorithms and the replacement policy affect the performance for wireless data access (i.e., the cache hit ratio and the transmission cost). We use a business-card service as the application for our wireless data access study. The business-card service is a generalization of the phone book feature in mobile handsets, which is one of the most popular features in mobile handsets. The business-card application will be elaborated in Section II.

II. Business-Card Application

Many applications such as mobile bank transactions and stock trades can utilize strongly consistent cache mechanisms. For purpose of illustration, this section uses a business-card application to study the performance for wireless data access. This application is similar to the address book feature built in most mobile handsets. Unlike the phone book, the business cards are stored and maintained in a business-card database in the network, and the most-frequently used (MFU) business cards are cached in the wireless terminal. This application offers four major advantages over the phone book feature in mobile handsets. First, when a user changes the handset, the phone book may not be conveniently transferred to the new handset. This is particularly true when the old handset is broken, and the phone book is lost. With our business-card service, the user can access his/her "private" phone book from any handset. Second, besides the private phone book, the business-card service can also maintain a "public" phone book database (just like yellow page) in the application server. Third, after a phone conversation, the business-card service allows the call parties to exchange their business cards that provide much more information than the phone numbers in the phone book feature. Finally, when the information (e.g., phone number) of a user is changed, it will not automatically reflect to the phone books of other users. With the business-card service, a user can update the business card in the database of the business-card

Manuscript received July 2, 2002; revised September 16, 2002; accepted October 1, 2002. The editor coordinating the review of this paper and approving it for publication is G. Cao. This work was supported in part by the MOE Program for Promoting Academic Excellence of Universities under Grant 89-E-FA04-1-4, by IIS, Academia Sinica, by FarEastone, and by the Lee and MTI Center for Networking Research, National Chiao Tung University.

Y.-B. Lin and J.-J. Chen are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, R.O.C. (e-mail: liny@liny.csie.nctu.edu.tw; chencz@pcs.csie.nctu.edu.tw).

W.-R. Lai is with the Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taiwan 320, R.O.C. (e-mail: wrlai@saturn.yzu.edu.tw).

Digital Object Identifier 10.1109/TWC.2003.819019

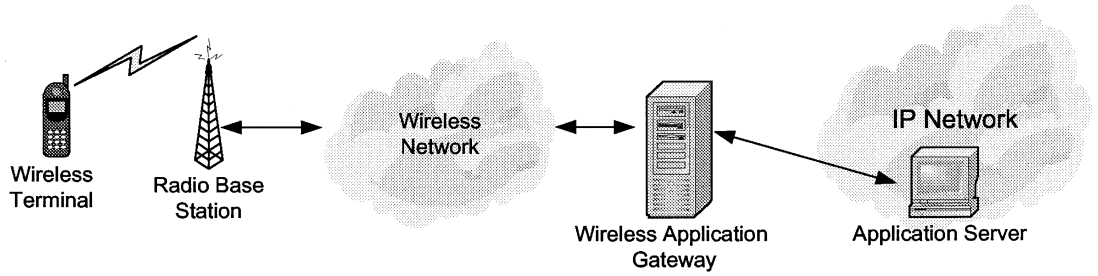

Fig. 1. Example of wireless application platform.

TABLE I
VCARD FORMAT IN THE iSMS-BASED BUSINESS CARD SERVICE

Field	Description	Length
VERSION	version of vCard	13 bytes
FN	vCard object name	30 bytes
N	name information	40 bytes
ORG	organization information	50 bytes
TITLE	job title	50 bytes
ADR	address	120 bytes
TEL	phone number	130 bytes
EMAIL	email address	50 bytes
URL	uniform resource locator	50 bytes
CALENDAR*	public calendar event	100 bytes

* The **CALENDAR** field is not defined in the vCard standard.

application server (typically through Internet), and other users always access the correct information.

We have implemented the business-card service in iSMS [15], a platform that integrates IP networks with the short message services (SMSs) in mobile telephone systems. iSMS provides a generic gateway for creating and hosting wireless data services for mobile stations (i.e., wireless terminals). The iSMS gateway is equivalent to the wireless application gateway in Fig. 1. This approach does not require any modification to the mobile telephone system architecture. Therefore, the iSMS system can be quickly developed and operated by a third party or an end user without involvement of mobile equipment manufacturers and telecom operators. In the iSMS-based business-card application, the format of the business card follows the vCard standard [5], as illustrated in Table I. In this format, the **FN** field is used to specify the vCard object. The **N** field is a single structured text value, which corresponds, in sequence, to the family name, given name, additional names, honorific prefixes, and honorific suffixes. A person may have several telephone numbers; e.g., work phone number, fax number, and cellular telephone number. In vCard, these numbers are included in the **TEL** field. We also include a **CALENDAR** field that allows the user to fill in the schedule of public events he/she wants to share with others. In our iSMS implementation, the size of a vCard can be unlimited in the iSMS application server. However, the vCard is tailored to be of fixed size when it is delivered to the wireless terminal. The length of each field in our implementation is shown in Table I. The appearance of an iSMS business card in PDA is illustrated in Fig. 2. An iSMS business card can be updated, added to or removed from the database by the application server or by the wireless terminal. The access of a business card is based on the algorithms described in Section III.

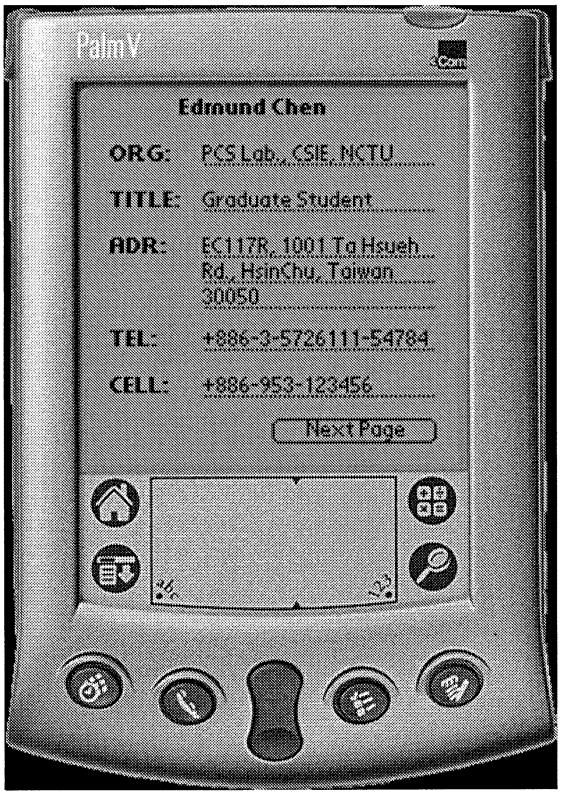

Fig. 2. iSMS business card shown in PDA.

III. CACHED DATA ACCESS AND REPLACEMENT ALGORITHMS

We consider two strongly consistent algorithms for wireless data access: poll-each-read (PER) and callback (CB) [25], [18]. In both algorithms, when a data object (a business card in our example) is updated at the wireless application server, the valid object is not immediately sent to update the cached copies in the wireless terminal. Instead, the valid object is delivered to a wireless terminal only when a data access operation is actually performed. Several time-to-live (TTL)-based algorithms were proposed for weakly consistent caching, which are out of the scope of this paper. For more details, the reader is referred to [11] and the reference therein. Cao [2], [3] proposed an IR-based cache invalidation algorithm, which efficiently utilizes the broadcast bandwidth to intelligently broadcast the data requested by clients. Kahol *et al.* [7] used synchronous invalidation reports to maintain cache consistency, i.e., reports are broadcast by their server only when some data changes. These

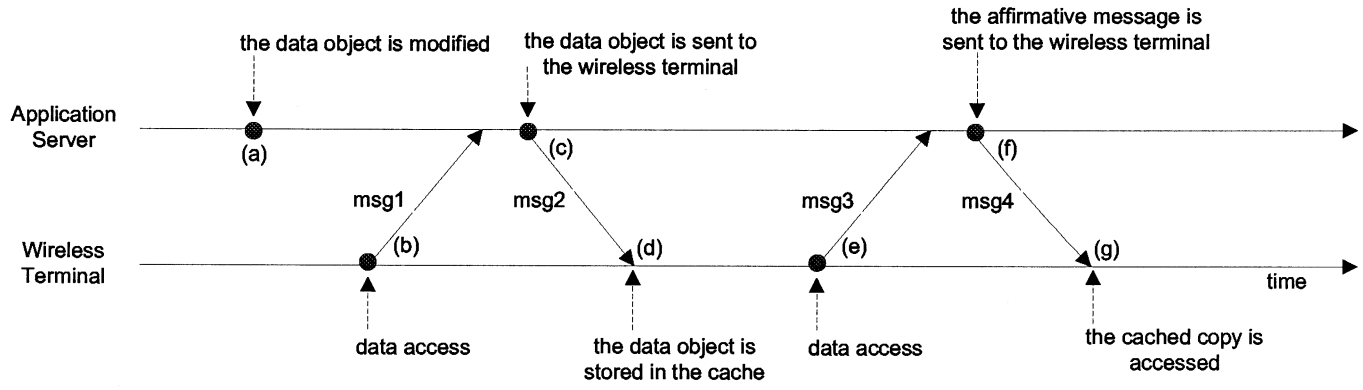

Fig. 3. Data access in PER.

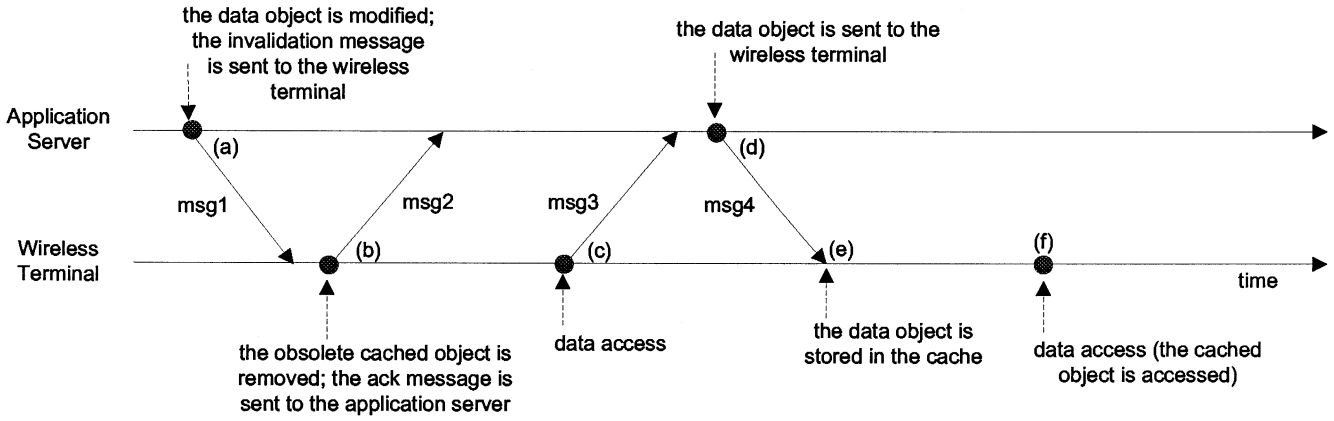

Fig. 4. Data access in CB.

schemes are implemented in lower layers of the wireless protocol stacks, which may perform well in wireless local area networks, but typically cannot be implemented in existing commercial mobile telecommunications networks such as the global system for mobile communication (GSM), general packet radio service, and universal mobile telecommunications systems. This paper considers PER and call-back that can be implemented at the application layer without modifying the core network protocols.

In PER [6], at a data access request [see Fig. 3(b) and (e)], the wireless terminal always asks the wireless application server to check if the cached object is valid. If so, the wireless application server responds affirmatively [Fig. 3(f)] and the user accesses the object in the cache of the wireless terminal [Fig. 3(g)]. If the data object is updated before the access [see Fig. 3(a) and (b)], the wireless application server sends the current data object to the wireless terminal [Fig. 3(c)], and this object is stored in the cache of the wireless terminal [Fig. 3(d)]. In this approach, when an object O_i is found in the cache, the wireless terminal still needs to obtain O_i from the wireless application server if O_i has been invalidated. Thus, a *cache hit* may not be beneficial to PER. Define *effective hit ratio* as the probability that for an access to object O_i, a cache hit occurs in the wireless terminal and the cached object is valid. In PER, the cost for accessing an object with effective cache hit is a cache affirmative request and acknowledgment exchange between the wireless application server and the wireless terminal (msg3 and msg4 in Fig. 3). For a cache miss or a cache hit where the data object is invalidated, the access cost is a request message sent from the wireless terminal (msg1 in Fig. 3) and a data object transmission from the wireless application server to the wireless terminal (msg2 in Fig. 3).

In the CB approach [6], [14], whenever a data object is modified, the wireless application server sends a message to invalidate the corresponding cached object in the wireless terminals [see Fig. 4(a)]. The cache storage of the invalidated object is reclaimed to accommodate other data objects, and the wireless terminal sends an acknowledgment to inform the application server that the invalidation is successful [see Fig. 4(b)]. During the period between Fig. 4(a) and (d), if other updates to this object occur, no invalidation message needs to be sent to the wireless terminal (because no invalidated copy will be found in the cache). In this approach, all objects stored in the cache are valid, and a cache hit is always an effective hit (if the message transmission delay is ignored). In a data access, if a cache hit occurs, the cached object is used without any communication between the application server and the wireless terminal [Fig. 4(f)]. For a cache miss, the access cost is a request message (msg3 in Fig. 4) sent from the wireless terminal and a data object transmission (msg4 in Fig. 4) from the application server to the wireless terminal. It is required to invalidate a cached object at the wireless terminal (if exists) when the object in the application server is updated (msg1 and msg2 in Fig. 4).

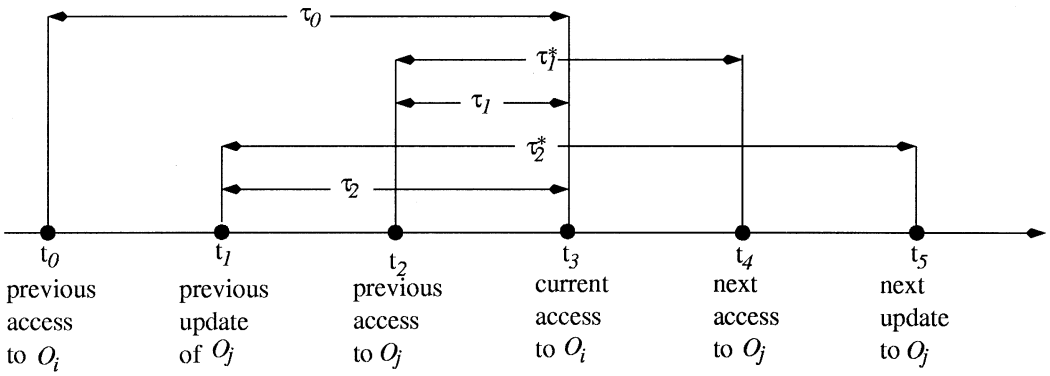

Fig. 5. Timing diagram I.

The typical cache size in a wireless terminal is not large. When the cache is full, some cached objects must be removed to accommodate new objects. We consider the *least-recently used* (LRU) replacement policy. This policy is often utilized to manage cache memory in computer architecture [20], virtual memory in operating systems [19], and location tracking in mobile phone networks [10]. LRU uses the recent past as an approximation of the near future, and replaces the cached object that has not been used for the longest period of time. LRU associates with each cached object the time of its last use. When a cached object must be replaced, LRU chooses the object that has not been used for the longest period of time. Therefore, every object in the cache has a rank. If a cached object has the Rank 1, it means that the object is most recently used. If an object O_i has a Rank k, it means that $k-1$ objects are more recently used than O_i is. If $k > K$, where K is the size of the cache, then O_i has been removed out of the cache.

The number of data objects in the application server is much larger than the cache size in the wireless terminal. However, our experience in exercising wireless applications [16] indicated that for an observed period, only a small number N of data objects in the wireless application server are potentially accessed by a wireless terminal. Although the objects to be accessed vary from time to time, the number N is not significantly larger than the cache size of the wireless terminal. That is, the data access pattern of a wireless terminal exhibits *temporal locality* [20], which is the tendency for a wireless terminal to access in the near future those data objects referenced in the recent past. Temporal locality may not be observed in wireline Internet access because the desktop users typically navigate through several web sites at the same time. On the other hand, the restricted user interface of wireless terminal only allows a user to access a small region of data objects for instant information acquisition. Thus, cache can effectively reduce the data access time for the wireless terminals. In Section IV, we propose an analytic model to investigate the cache performance for the wireless terminals.

IV. Analytic Modeling

This section proposes analytic models for the business-card application based on PER and CB. Note that our model is applicable to other general remote caching scenarios. We first model the wireless data access when LRU is exercised with PER. To simplify our discussion, we assume that the sizes of data objects are the same. This assumption is justified for the iSMS business-card application where the size of a vCard is fixed. Consider a data object O_i in the cache of a wireless terminal. Based on the LRU policy, when O_i is accessed, it is moved to the top of the cache. Between the previous and the next accesses to O_i, if the wireless terminal accesses to k distinct data objects other than O_i, then O_i is moved to the $k+1$th position of the cache. If $k = K$, then O_i is removed from the cache before the next access to O_i arrives. In this case, a *cache miss* to O_i occurs. Consider the timing diagram in Fig. 5, where the current access to O_i occurs at time t_3. If the previous access to O_i occurs at time t_0, then the interdata access time to O_i is $\tau_0 = t_3 - t_0$. Assume that accesses to a data object are a Poisson process with rate μ_i. In this section, we assume that $\mu_i = \mu_j = \mu$ for $1 \le i, j \le N$. In other words, the access patterns for all objects are the same. In our simulation experiments, this assumption will be relaxed (see Sections V and VI). Therefore, τ_0 is exponentially distributed with mean $1/\mu$. Consider the accesses to another data object O_j. After the access at t_0, O_i is stored in the cache with Rank 1. Suppose that the last access to O_j before t_3 occurs at t_2, and the first access to O_j after t_3 occurs at t_4. Then, $\tau_1^* = t_4 - t_2$ has the same distribution as τ_0. Furthermore, data access to O_i at t_3 is a random observer to τ_1^* (property of the Poisson process). From the excess life theorem [17] and the memoryless property of the exponential distribution, the period $\tau_1 = t_3 - t_2$ has the same distribution as τ_1^*. If $t_0 < t_2$ (i.e., $\tau_1 < \tau_0$), then the access to O_j will increment the rank of O_i by one before the access to O_i occurs at time t_3. Let $\theta_1(t) = \Pr[\tau_1 < t]$. Then

$$\theta_1(t) = \int_{\tau_1=0}^{t} \mu e^{-\mu\tau_1} d\tau_1 = 1 - e^{-\mu t}. \tag{1}$$

Let $\theta_2(t,k)dt$ be the probability that $\tau_0 \in [t, t+dt]$ and O_i's rank is increased to $k+1$ in the period τ_0. In other words, k distinct data objects other than O_i are accessed during τ_0. For $0 \le k < N$, from (1), we have

$$\begin{aligned}\theta_2(t,k) &= \mu e^{-\mu t}\binom{N-1}{k}\Big[\theta_1(t)\Big]^k\Big[1-\theta_1(t)\Big]^{N-k-1} \\ &= \mu\binom{N-1}{k}\left[\sum_{l=0}^{k}(-1)^l\binom{k}{l}e^{-\mu(N+l-k)t}\right]. \end{aligned} \tag{2}$$

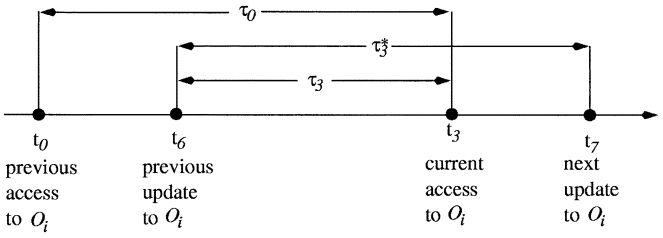

Fig. 6. Timing diagram II.

When the wireless terminal accesses O_i at time t_3, the cached copy of O_i may have already been obsolete due to updates to the original copy at the wireless application server. Consider the timing diagram in Fig. 6. Suppose that the last update to O_i before t_3 occurs at t_6, and the first update to O_i after t_3 occurs at t_7. Assume that the interupdate time $\tau_3^* = t_7 - t_6$ has a general distribution with the density function $f(\tau_3^*)$, the mean $1/\lambda$, and the Laplace transform

$$f^*(s) = \int_{\tau_3^*=0}^{\infty} f(\tau_3^*) e^{-\tau_3^* s} d\tau_3^*.$$

Since the accesses to O_i are a Poisson stream, t_3 is a random observer of τ_3^*. From the excess life theorem [17], $\tau_3 = t_3 - t_6$ has the density function

$$r(\tau_3) = \lambda \int_{\tau=\tau_3}^{\infty} f(\tau) d\tau$$

the distribution $R(\tau_3)$, and the Laplace transform

$$r^*(s) = \left(\frac{\lambda}{s}\right)\left[1 - f^*(s)\right]. \tag{3}$$

In PER, the cached copy of O_i is not invalidated during the period τ_0 if $\tau_3 > \tau_0$. Let $p_{\rm PER}$ be the *effective cache hit ratio* or the probability that when LRU is exercised with PER, the cached copy of O_i exists and is not obsolete. Suppose that the cache size of the wireless terminal is K and N (where $N \geq K$) is the number of objects in the wireless application server. Suppose that k distinct objects other than O_i are accessed during τ_0. The cached copy of O_i still exists in the cache at time t_3 if $k < K$. Therefore, from (2), we have

$$\begin{aligned} p_{\rm PER} &= \sum_{k=0}^{K-1} \int_{\tau_0=0}^{\infty} \int_{\tau_3=\tau_0}^{\infty} \theta_2(\tau_0, k) r(\tau_3) d\tau_3 d\tau_0 \qquad (4) \\ &= \sum_{k=0}^{K-1} \int_{\tau_0=0}^{\infty} \theta_2(\tau_0, k)\left[1 - R(\tau_0)\right] d\tau_0 \\ &= \sum_{k=0}^{K-1} \binom{N-1}{k} A_1 \\ &= \sum_{k=0}^{K-1} \binom{N-1}{k} A_2 \qquad (5) \end{aligned}$$

where

$$A_1 = \sum_{l=0}^{k} (-1)^l \binom{k}{l} \int_{\tau_0=0}^{\infty} \mu\left[1 - R(\tau_0)\right] e^{-\mu(N+l-k)\tau_0} d\tau_0$$

and

$$A_2 = \sum_{l=0}^{k} \frac{(-1)^l \binom{k}{l}\left[1 - r^*(\mu(N+l-k))\right]}{N+l-k}.$$

Using (3), (5) can be computed numerically by a computer program. If τ_3^* is exponentially distributed, then τ_3 is also exponentially distributed with the same mean $1/\lambda$. In this case, (4) can be expressed as (6), shown at the bottom of the page. Let $x = e^{-\mu\tau_0}$. Then, we have

$$d\tau_0 = \frac{-dx}{\mu x}$$

and (6) is rewritten as

$$\begin{aligned} p_{\rm PER} &= \sum_{k=0}^{K-1} \binom{N-1}{k} \int_{x=0}^{1} x^{N-k+\lambda/\mu-1}\left(1-x\right)^k dx \\ &= \sum_{k=0}^{K-1} \binom{N-1}{k} \left[\frac{k!}{\prod_{m=0}^{k}\left(N-k+\frac{\lambda}{\mu}+m\right)}\right]. \qquad (7) \end{aligned}$$

When LRU is exercised with CB, the cache position ranking of O_i is not affected by an access to another object O_j if O_j is invalidated after the access. Consider the timing diagram in Fig. 5. Suppose that the previous update to O_j before t_3 occurs at t_1, and the first update to O_j after t_3 occurs at t_5. As we discussed before, the interupdate time $\tau_2^* = t_5 - t_1$ has the density function $f(\tau_2^*)$, and the period τ_2 has the density function $r(\tau_2)$. Under CB, the updates to O_j during τ_0 will affect the ranking of the cached entry for O_i if the last access to O_j occurs at t_2 and the last update to O_j occurs at t_1, where $t_2 > t_1$ (i.e., $\tau_1 < \tau_2$). Let $\theta_3(t)$ be the probability that the above situation occurs under the condition that $\tau_0 = t$. We have

$$\begin{aligned} \theta_3(t) &= \Pr[\tau_1 < \tau_0, \tau_1 < \tau_2 | \tau_0 = t] \\ &= \int_{\tau_1=0}^{t} \int_{\tau_2=\tau_1}^{\infty} \mu e^{-\mu\tau_1} r(\tau_2) d\tau_2 d\tau_1 \\ &= \int_{\tau_1=0}^{t} \left[1 - R(\tau_1)\right] \mu e^{-\mu\tau_1} d\tau_1 \\ &= 1 - e^{-\mu t} - \int_{\tau_1=0}^{t} \mu e^{-\mu\tau_1} R(\tau_1) d\tau_1. \qquad (8) \end{aligned}$$

Let $\theta_4(t,k)dt$ be the probability that k data objects other than O_i will have smaller ranks (higher priorities) in the cache during τ_0 when $\tau_0 \in [t, t+dt]$. Therefore

$$\theta_4(t,k) = \mu e^{-\mu t} \binom{N-1}{k} \left[\theta_3(t)\right]^k \left[1 - \theta_3(t)\right]^{N-k-1}. \tag{9}$$

$$\begin{aligned} p_{\rm PER} &= \sum_{k=0}^{K-1} \binom{N-1}{k} \int_{\tau_0=0}^{\infty} \mu e^{-\mu\tau_0} \left[\int_{\tau_3=\tau_0}^{\infty} \lambda e^{-\lambda\tau_3} d\tau_3\right] \times \left(1 - e^{-\mu\tau_0}\right)^k \left(e^{-\mu\tau_0}\right)^{N-k-1} d\tau_0 \\ &= \sum_{k=0}^{K-1} \binom{N-1}{k} \int_{\tau_0=0}^{\infty} \mu e^{-(\mu+\lambda)\tau_0} \left(1 - e^{-\mu\tau_0}\right)^k \times \left(e^{-\mu\tau_0}\right)^{N-k-1} d\tau_0 \qquad (6) \end{aligned}$$

Let $p_{\rm CB}$ be the probability that when CB is exercised, an effective cache hit occurs at time t_3. From (9) we have (10), shown at the bottom of the page. If τ_3^* is exponentially distributed, then (8) is rewritten as

$$\theta_3(t) = \left(\frac{\mu}{\mu+\lambda}\right)\left[1 - e^{-(\mu+\lambda)t}\right]$$

and (9) is rewritten as

$$\theta_4(t,k) = \mu e^{-\mu t}\binom{N-1}{k}\left\{\left(\frac{\mu}{\mu+\lambda}\right)\left[1-e^{-(\mu+\lambda)t}\right]\right\}^k \times\left\{1-\left(\frac{\mu}{\mu+\lambda}\right)\left[1-e^{-(\mu+\lambda)t}\right]\right\}^{N-k-1}.$$

Therefore, (10) is derived as (11), shown at the bottom of the page. Let $x = e^{-(\mu+\lambda)\tau_0}$. Then

$$d\tau_0 = \frac{-dx}{(\mu+\lambda)x}. \tag{12}$$

From (12), (11) is rewritten as (13), shown at the bottom of the page. Let $y = (\mu/(\mu+\lambda))(1-x)$, we have $dx = -((\mu+\lambda)/\mu)\,dy$, and (13) is rewritten as (14) shown at the bottom of the page. Let $l = m + k$, (14) is rewritten as

$$p_{\rm CB} = \left[\frac{1}{N(\mu+\lambda)^N}\right]\left[\sum_{k=0}^{K-1}\sum_{l=k+1}^{N}\binom{N}{l}\lambda^{N-l}\mu^l\right]. \tag{15}$$

The probabilities $p_{\rm PER}$ and $p_{\rm CB}$ will be used to compute the wireless data access costs that consist of the following components:

- $c_{\rm req}$: the transmission cost of a data access request message (e.g., msg1 and msg3 in Fig. 3, and msg3 in Fig. 4);
- $c_{\rm ack}$: the transmission cost of an affirmative message in PER (i.e., msg4 in Fig. 3) or an invalidation acknowledgment message in CB (i.e., msg2 in Fig. 4);
- $c_{\rm inv}$: the transmission cost of an invalidation message in CB (i.e., msg1 in Fig. 4);
- $c_{\rm obj}$: the transmission cost of a data object (i.e., msg2 in Fig. 3 and msg4 in Fig. 4).

Consider a long time interval. In this interval, N_a data accesses are observed. For CB, $N_{\rm inv}$ invalidation messages are sent.

$$\begin{aligned} p_{\rm CB} &= \sum_{k=0}^{K-1}\int_{\tau_0=0}^{\infty}\int_{\tau_3=\tau_0}^{\infty}\theta_4(\tau_0,k)r(\tau_3)d\tau_3 d\tau_0 \\ &= \sum_{k=0}^{K-1}\binom{N-1}{k}\int_{\tau_0=0}^{\infty}\mu e^{-\mu\tau_0}\Big[\theta_3(\tau_0)\Big]^k \times\Big[1-\theta_3(\tau_0)\Big]^{N-k-1}\Big[1-R(\tau_0)\Big]d\tau_0 \end{aligned} \tag{10}$$

$$p_{\rm CB} = \sum_{k=0}^{K-1}\binom{N-1}{k}\int_{\tau_0=0}^{\infty}\mu e^{-(\mu+\lambda)\tau_0}\times\left\{\left(\frac{\mu}{\mu+\lambda}\right)\left[1-e^{-(\mu+\lambda)\tau_0}\right]\right\}^k\times\left\{1-\left(\frac{\mu}{\mu+\lambda}\right)\left[1-e^{-(\mu+\lambda)\tau_0}\right]\right\}^{N-k-1}d\tau_0 \tag{11}$$

$$p_{\rm CB} = \sum_{k=0}^{K-1}\binom{N-1}{k}\int_{x=0}^{1}\left(\frac{\mu}{\mu+\lambda}\right)\left[\left(\frac{\mu}{\mu+\lambda}\right)(1-x)\right]^k\times\left[1-\left(\frac{\mu}{\mu+\lambda}\right)(1-x)\right]^{N-k-1}dx \tag{13}$$

$$\begin{aligned} &p_{\rm CB} \\ &= \sum_{k=0}^{K-1}\binom{N-1}{k}\int_{y=0}^{\mu/\mu+\lambda} y^k(1-y)^{N-k-1}dy \\ &= \left(\frac{1}{\mu+\lambda}\right)^N \\ &\quad\cdot\left\{\sum_{k=0}^{K-1}\binom{N-1}{k}\times\left[\frac{\lambda^{N-k-1}\mu^{k+1}}{k+1}+\frac{(N-k-1)\lambda^{N-k-2}\mu^{k+2}}{(k+1)(k+2)}+\frac{(N-k-1)(N-k-2)\lambda^{N-k-3}\mu^{k+3}}{(k+1)(k+2)(k+3)}+\cdots+\frac{k!(N-k-1)!\mu^N}{N!}\right]\right\} \\ &= \sum_{k=0}^{K-1}\sum_{m=1}^{N-k}\binom{N-1}{k+m-1}\left[\frac{\lambda^{N-k-m}\mu^{k+m}}{(k+m)(\mu+\lambda)^N}\right] \end{aligned} \tag{14}$$

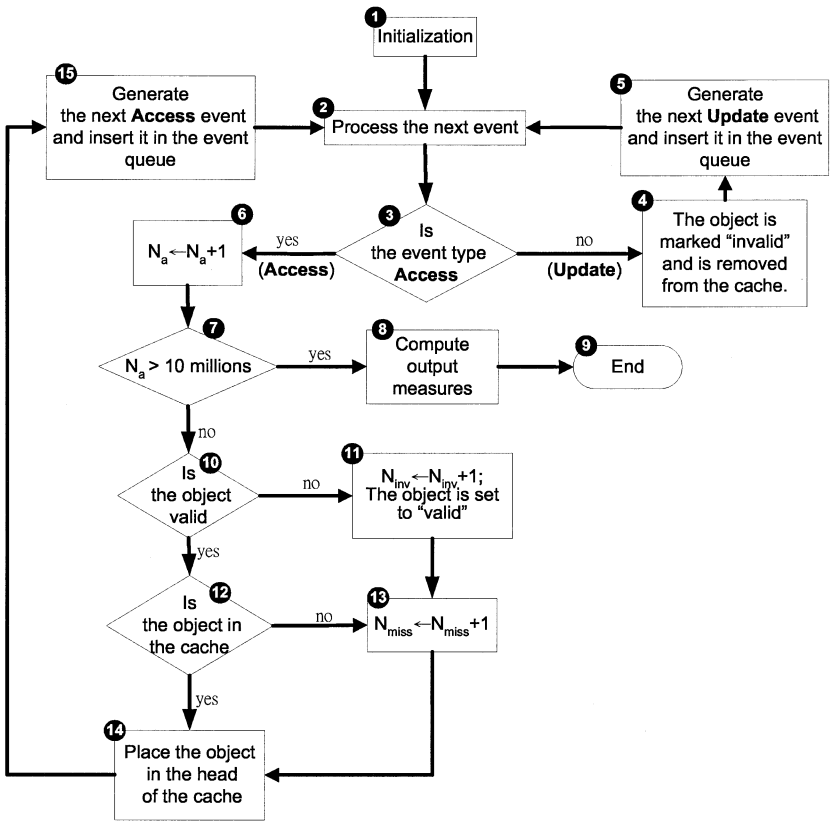

Fig. 7. Simulation flowchart for CB.

From Figs. 3 and 4, the net wireless transmission costs of PER and CB (during the observation period) are expressed as

$$C^*_{\rm PER} = N_a\left[c_{\rm req} + p_{\rm PER}c_{\rm ack} + (1-p_{\rm PER})\,c_{\rm obj}\right] \quad (16)$$

$$C^*_{\rm CB} = N_{\rm inv}(c_{\rm inv}+c_{\rm ack}) + N_a(1-p_{\rm CB})(c_{\rm req}+c_{\rm obj}). \quad (17)$$

Consider the per access cost $C_{\rm PER}$ for PER. From (16)

$$C_{\rm PER} = \frac{C^*_{\rm PER}}{N_a} = c_{\rm req} + p_{\rm PER}c_{\rm ack} + (1-p_{\rm PER})\,c_{\rm obj}. \quad (18)$$

From (17), the per access cost $C_{\rm CB}$ for CB can be expressed as

$$C_{\rm CB} = \frac{C^*_{\rm CB}}{N_a} = p_{\rm inv}(c_{\rm inv}+c_{\rm ack}) + (1-p_{\rm CB})\,(c_{\rm req}+c_{\rm obj}) \quad (19)$$

where $p_{\rm inv} = N_{\rm inv}/N_a$ is the probability that for a data object, one or more updates to this object occurring during the period between two data accesses; that is, the probability that $\tau_3 < \tau_0$ in Fig. 6. Therefore

$$\begin{aligned} p_{\rm inv} &= \Pr[\tau_3 < \tau_0] \\ &= \int_{\tau_3=0}^{\infty}\int_{\tau_0=\tau_3}^{\infty} r(\tau_3)\mu e^{-\mu\tau_0}\,d\tau_0 d\tau_3 \\ &= \left(\frac{\lambda}{\mu}\right)\left[1-f^*(\mu)\right] \end{aligned} \quad (20)$$

which is the same as (3) where $s=\mu$.

V. Simulation Model

We utilize discrete event simulation to model wireless data access. In the simulation two types of events are defined: **Access** and **Update**. There are N data objects in the application server and the cache size in a wireless terminal is K. The inter-**Access** event arrival times for an object O_i are drawn from an exponential distribution with rate μ_i, based on (23), which will be elaborated in Section VI. Note that the exponential restriction can be easily relaxed in our simulation model. The inter-**Update** event arrival times for O_i are drawn from a Gamma distribution with mean $1/\lambda$ and the variance ν. We are particularly interested in the Gamma distribution. It has been shown that the distribution of any positive random variable can be approximated by a mixture of Gamma distributions (see [8, Lemma 3.9]). When $\nu = 1/\lambda^2$, the Gamma distribution becomes an exponential distribution, which can be used to provide the mean value analysis [9]. It suffices to use the Gamma distributions with different shape and scale parameters to represent different interupdate arrival time distributions [13]. We first describe the CB simulation. The output measures of the simulation are N_a (the number of data accesses), $N_{\rm miss}$ (the number of data accesses that result in data object transmissions from the application server to the wireless terminal), and $N_{\rm inv}$ (the number of invalidations). In the CB simulation, these output measures are used to compute $p_{\rm CB}$ and $p_{\rm inv}$ as follows:

$$p_{\rm CB} = 1 - \frac{N_{\rm miss}}{N_a} \quad \text{and} \quad p_{\rm inv} = \frac{N_{\rm inv}}{N_a}. \quad (21)$$

The flowchart of the CB simulation is shown in Fig. 7, and is described as follows.

Step 1. Initially, all data objects are marked *valid*. The first **Update** and **Access**

events are generated. The timestamps of these events are computed based on the interarrival time distributions previously mentioned. The events are inserted in the event list in the nondecreasing timestamp order.

Step 2. The event e at the head of the event list is processed.

Step 3. If the event type of e is **Access**, then Step 6 is executed. Otherwise Step 4 is executed.

Step 4. The updated object is marked *invalid*. For CB, if the object is in the cache, the cache entry is purged.

Step 5. The next **Update** event is generated and inserted in the event list.

Step 6. If the event type of e is **Access** at Step 3, then N_a is incremented by one.

Step 7. If 10 000 000 **Access** events have been processed, then Step 8 is executed. Otherwise, the simulation proceeds to Step 10. In our simulation experiments, the confidence intervals of the 99% confidence levels are within 3% of the mean values in most cases.

Steps 8 and 9. The output measures $p_{\rm CB}$ and $p_{\rm inv}$ are computed by using (21), and the cost $C_{\rm CB}$ can be computed by using (21) and (19). Then the simulation run terminates.

Steps 10–14. If the cached object is *invalid* at Step 10, then $N_{\rm inv}$ is incremented by one. The valid copy is sent from the application server to the cache, and the cached object is set to *valid* at Step 11. In this case, the object is obtained from the application server and $N_{\rm miss}$ is incremented by one at Step 13. The object is placed in the head of the cache according to the LRU policy at Step 14. Then the simulation flow proceeds to Step 15. If the object is *valid*, then two cases are considered at Step 12. If the object is in the cache, then it is an effective cache hit, and the flow jumps to Step 14. If the object is not in the cache at Step 12, it implies that the object has been removed from the cache before it is accessed again. Steps 13 and 14 are executed.

Step 15. The next **Access** event is generated and inserted in the event list.

The simulation flowchart for PER is similar to that in Fig. 7 except that at Step 4, the invalid object is not removed from the cache. Also the variable $N_{\rm inv}$ at Step 11 is not needed in the PER simulation.

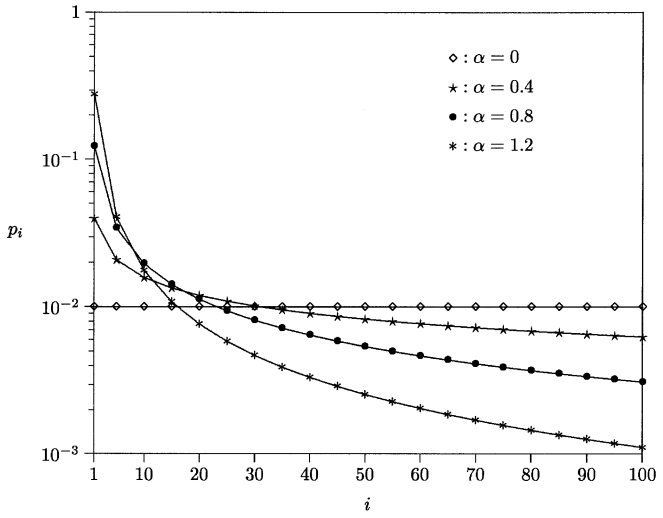

Fig. 8. Cutoff Zipf-like distribution ($N = 100$).

Note that Step 14 can be modified to simulate different cache replacement policies. In [4], similar results were observed for policies including LRU, MFU, First In First Out, Second Chance, and TTL. This paper only considers LRU. The effects of other replacement policies are out of the scope of this paper, and the reader is referred to [4] and the references therein.

VI. Numerical Examples

This section uses the business-card service to illustrate the performance of wireless data access. Our study assumes that the relative frequency with which the data objects are accessed follows a generalization of Zipf's law [1], [26]. Let the N data objects (of the application server) be ranked in order of their popularity where O_i is the ith most popular object. Let p_i be the probability that an incoming access is for O_i. If p_i has a "cutoff" Zipf-like distribution, then for $1 \le i \le N$

$$p_i = \frac{\Omega}{i^{\alpha}}, \qquad \text{where} \quad \Omega = \left(\sum_{i=1}^{N} \frac{1}{i^{\alpha}} \right)^{-1}. \tag{22}$$

If the net access rate to data objects is μ^*, then from (22), the data access rate to O_i is

$$\mu_i = p_i \mu^* = \frac{\Omega \mu^*}{i^{\alpha}}. \tag{23}$$

Fig. 8 plots p_i as a function of α. It is clear that the larger the α value, the better the temporal locality. In [1], the α values are reported to be in the range of 0.64–0.98. When $\alpha = 0$, (22) is simplified as $p_i = 1/N$ for all i, and the data access rates to all objects are the same, which is the situation we considered in Section IV. Table II shows that for $\alpha = 0$, the analytic and simulation models are consistent. [In this table, ρ is defined in (24)]. Therefore, both the analytic model and the simulation implementation are validated. Since the cache performance is better for a larger α value, the analytic analysis in Section IV (where $\alpha = 0$) can also be considered as a worst case analysis. For $\alpha \neq 0$, the derivations in the analytic model will become very tedious. In this case, we investigate the performance of wireless data access through the simulation model.

TABLE II
THE p_{PER} AND p_{CB} VALUES: ANALYTIC ANALYSIS VERSUS SIMULATION ($N = 100$, $K = 50$, ff$\nu = 1/\lambda^2$). (a) THE EFFECTIVE HIT RATIO FOR PER. (b) THE EFFECTIVE HIT RATIO FOR CB

ρ	2	4	6	8	10
Simulation	43.00%	46.31%	47.47%	48.10%	48.53%
Analytic	43.01%	46.31%	47.50%	48.10%	48.48%
Error	0.02%	0%	0.06%	0%	0.10%

(a)

ρ	2	4	6	8	10
Simulation	49.15%	49.67%	49.78%	49.84%	49.94%
Analytic	50%	50%	50%	50%	50%
Error	1.70%	0.66%	0.44%	0.32%	0.12%

(b)

TABLE III
THE p_{inv} VALUES: ANALYTIC ANALYSIS VERSUS SIMULATION ($N = 100$,$\alpha = 0.8$)

ν (Unit: $1/\lambda^2$)	0.01	0.1	1	10	100
Simulation	0.9971%	0.9996%	0.9713%	0.4693%	0.0826%
Analytic	1%	1%	0.9822%	0.477%	0.0843%
Error	0.29%	0.04%	1.11%	1.61%	1.93%

(a) $\rho = 100$

ν (Unit: $1/\lambda^2$)	0.01	0.1	1	10	100
Simulation	48.7083%	47.7833%	40.3783%	18.4805%	4.1272%
Analytic	48.7043%	47.7697%	40.3135%	18.4877%	4.1318%
Error	0.01%	0.03%	0.16%	0.04%	0.11%

(b) $\rho = 1$

ν (Unit: $1/\lambda^2$)	0.01	0.1	1	10	100
Simulation	86.7696%	86.0791%	80.1136%	55.1635%	20.3659%
Analytic	86.8169%	86.0839%	80.0806%	55.2317%	20.4229%
Error	0.05%	0.01%	0.04%	0.12%	0.28%

(c) $\rho = 0.1$

As mentioned in Section V, we assume that the inter-**Update** arrival times for O_i are drawn from a Gamma distribution with mean $1/\lambda$ and the variance ν. The variance ν may have significant impact on p_{inv}. By varying ν, we compare the p_{inv} values of simulation and (20). Table III indicates that the analytic and the simulation results are consistent. In this table, the parameter ρ is defined as follows. Since we assume that the update rate to a data object is λ, the net update rate to the business-card database is $N\lambda$, and the access-to-update ratio for the whole system can be defined as

$$\rho = \frac{\sum_{i=1}^{N} \mu_i}{N\lambda} = \frac{\mu^*}{N\lambda}. \tag{24}$$

In the business-card application, the standard vCard information is seldom modified. On the other hand, the **CALENDAR** field may be frequently updated. Therefore, ρ may range from 0.01 (e.g., the **CALENDAR** field is updated daily) to 100 (e.g., the **CALENDAR** field is not provided and the address information is infrequently updated).

In iSMS, the information exchanged between the application server and the wireless terminal is delivered through short messages. The maximum size of the user data in a GSM short message is 140 B [12]. By including the SMS header overhead and considering the maximum size of the user data field, we have

$$c_{\mathrm{req}} = c_{\mathrm{ack}} = c_{\mathrm{inv}} = 45 \text{ B and } c_{\mathrm{obj}} = 727 \text{ B}. \tag{25}$$

Note that the access request, invalidation, invalidation acknowledgment, and affirmative messages are delivered by single short messages. The vCard transmission is done via multiple messages using the SMS concatenation technique. Also note that the uplink transmission cost may be different from the downlink transmission cost. For simplicity, we assume the uplink and downlink transmission costs are the same.

Based on the above discussion, we investigate the effects of K, ρ, α and ν as follows. This study assumes $N = 100$. For other N values, similar results are observed and will not be presented.

A. Effects of K and ρ

Fig. 9 plots p_{PER} and p_{CB} against ρ with various K values. When ρ is small (e.g., $\rho < 0.2$), it is likely that a data object has been updated before the next access arrives. Therefore, the cache performance is dominated by the update frequency, and the cache sizes only insignificantly affect the effective hit ratios. When ρ is large (e.g., $\rho > 10$), updates occur infrequently, which do not have significant impact on the effective hit ratio. Therefore, the effective hit ratio is only affected by the cache size. In this case, if we increase the cache size from $0.3N$ to $0.4N$, the effective hit ratio is improved by 20%. If the cache size is increased from $0.3N$ to $0.5N$, the improvement is 30%.

The $C_{\mathrm{PER}}/C_{\mathrm{CB}}$ curves in Fig. 9 show an interesting phenomenon that when $p_{\mathrm{PER}} = p_{\mathrm{CB}}$, $C_{\mathrm{PER}} < C_{\mathrm{CB}}$ for a small ρ, and $C_{\mathrm{PER}} > C_{\mathrm{CB}}$ for a large ρ. This phenomenon can be explained by the fact that p_{inv} decreases as ρ increases (see Table III). Let $p_{\mathrm{PER}} = p_{\mathrm{CB}}$, and the wireless transmission costs satisfy (25). Then from (18) and (19), we have

$$C_{\mathrm{PER}} > C_{\mathrm{CB}}, \qquad \text{if} \quad p_{\mathrm{inv}} < p_{\mathrm{PER}}; \qquad \text{where } p_{\mathrm{PER}} = p_{\mathrm{CB}}.$$

B. Effects of α

Fig. 10 indicates that the effective hit ratio increases as α increases. The effect of α is significant when $\alpha > 0.4$. The slopes of the α curves are relatively flat when $\alpha < 0.4$, which implies that the change of α only has insignificant impact on the effective hit ratios. In other words, when $\alpha < 0.4$, not enough temporal locality can be explored to increase the effective hit ratio. The $C_{\mathrm{PER}}/C_{\mathrm{CB}}$ curves in Fig. 10 shows similar results as that in Fig. 9: $C_{\mathrm{PER}} < C_{\mathrm{CB}}$ when ρ is small, and $C_{\mathrm{PER}} > C_{\mathrm{CB}}$ when ρ is large, which is independent of the α values. When $\rho = 1$, we observe that when α is small, $C_{\mathrm{PER}} < C_{\mathrm{CB}}$. When α is large, $C_{\mathrm{PER}} > C_{\mathrm{CB}}$.

C. Effects of the Object Size

Fig. 11 indicates that PER is more likely to outperform CB when the object size is small. For example, when $\alpha = 0.61$, $C_{\mathrm{CB}} > C_{\mathrm{PER}}$ for $c_{\mathrm{obj}} = 100$, and the result reverses for $c_{\mathrm{obj}} = 727$.

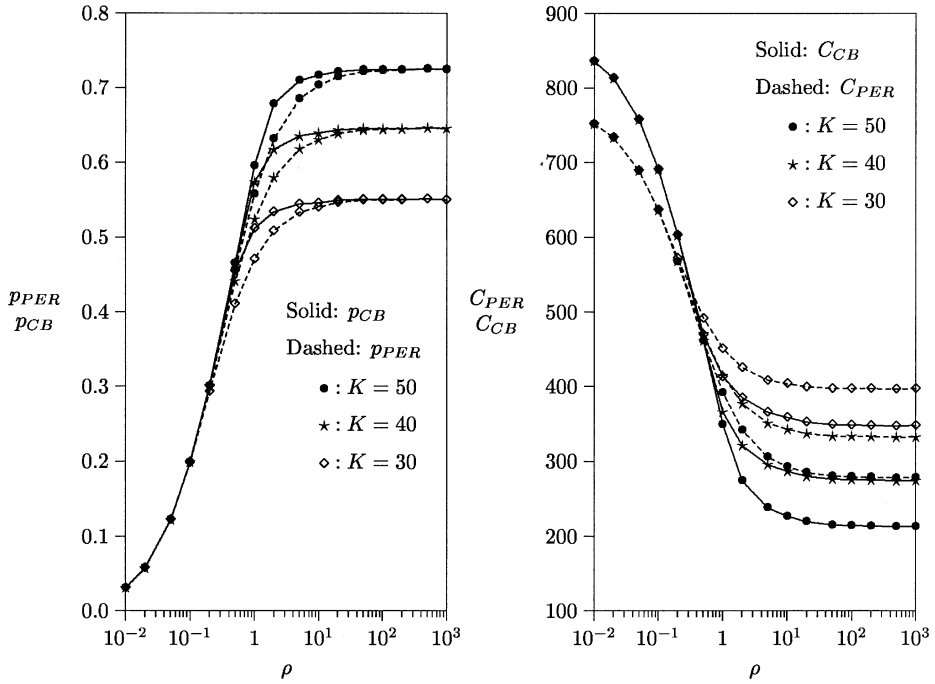

Fig. 9. Effects of K and ρ ($N = 100, \alpha = 0.8, \nu = 1/\lambda^2$).

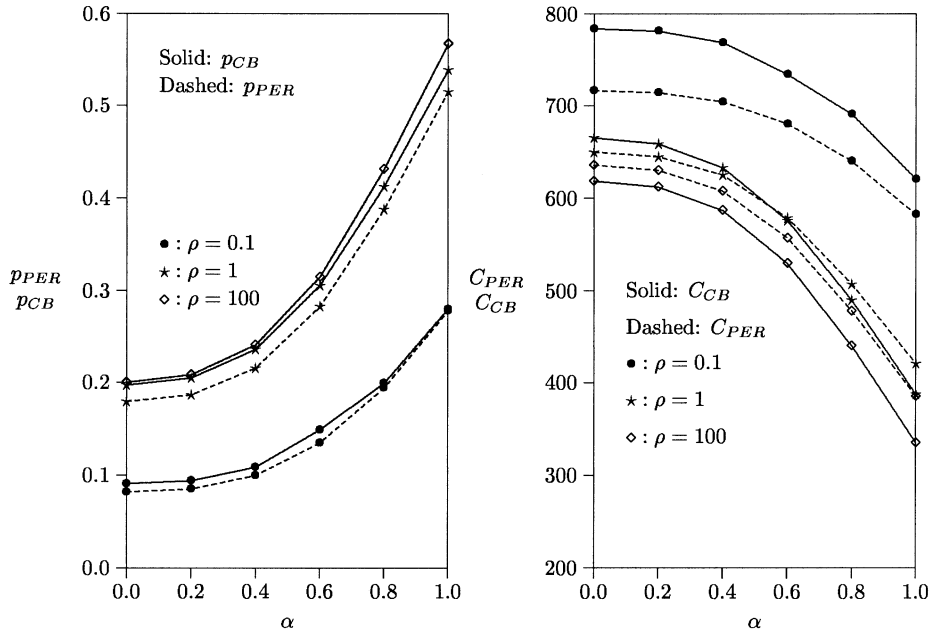

Fig. 10. Effects of α ($N = 100, K = 20, \nu = 1/\lambda^2$).

D. Effects of ν

Fig. 12 shows that the effective hit ratio increases as the variance ν of the interdata update intervals increases. When ν increases, more long and short interdata update intervals are observed. That is, it is likely to either find many updates occurring between two data accesses or no update is found between the two accesses. Therefore, as ν increases, p_{inv} decreases (see Table III) and the effective hit ratio increases. This effect is significant when ρ is small (when the update frequency is high).

The $C_{\text{PER}}/C_{\text{CB}}$ curves in Fig. 12 indicate that for $\rho > 1$, $C_{\text{PER}} > C_{\text{CB}}$ no matter how ν changes. For $\rho = 0.1$, $C_{\text{PER}} < C_{\text{CB}}$ when ν is small, and $C_{\text{PER}} > C_{\text{CB}}$ when ν is large. This phenomenon can be explained by (18) and (19) and the fact that p_{inv} decreases as ν increases.

E. Effects of PER and CB on the Effective Hit Ratio

It is clear that the inequality $p_{\text{CB}} > p_{\text{PER}}$ always holds because invalid data objects may exist in the cache when LRU is

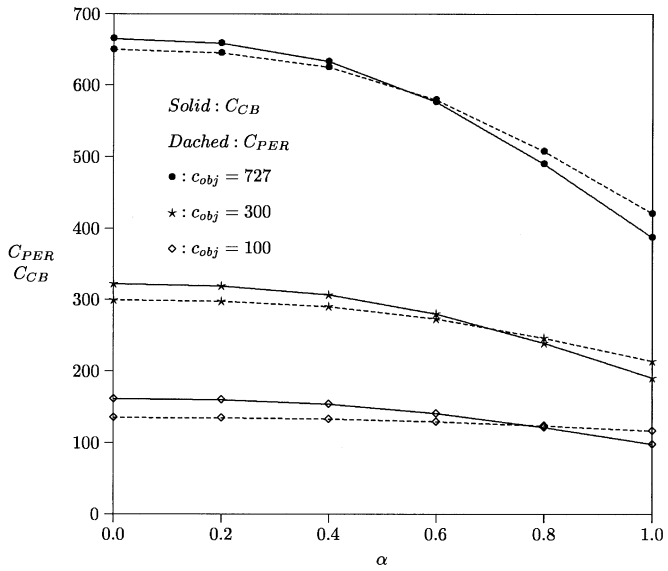

Fig. 11. Effects of the object size ($N = 100, K = 20, \rho = 1, \nu = 1/\lambda^2$).

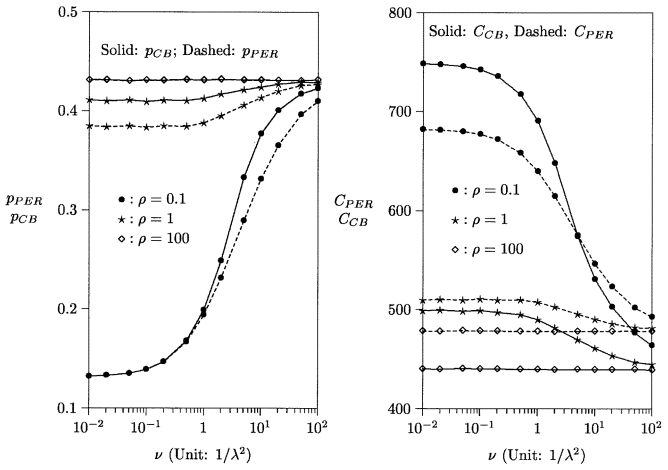

Fig. 12. Effects of ν ($N = 100, K = 20, \alpha = 0.8$).

exercised with PER. In this case, valid data objects may be removed from the cache while the invalid objects are kept in the cache. Fig. 9 indicates that both $p_{\rm PER}$ and $p_{\rm CB}$ have similar performance when $\rho > 10$ or $\rho < 0.1$. As we mentioned before, when ρ is large, the effective hit ratio is only affected by the cache size, and when ρ is small, the effective hit ratio is only affected by the data update frequency. Therefore, the CB and PER protocols have different effects on the hit ratios only when $0.1 < \rho < 100$. Fig. 10 indicates that the discrepancy between $p_{\rm PER}$ and $p_{\rm CB}$ becomes insignificant when α is large. In this case, temporal locality is high, and no matter which data access protocol is exercised, only a few objects are accessed. These accesses are likely to be cache hits. Fig. 12 shows that when $\rho = 0.1$, the discrepancy between $p_{\rm PER}$ and $p_{\rm CB}$ becomes significant as ν increases. On the other hand, when $\rho = 1$, this discrepancy becomes less significant as ν increases. These opposite phenomena are due to complicated interactions between ρ and $p_{\rm inv}$.

Let $\Delta_{\rm PER}$ and $\Delta_{\rm CB}$ be the "cost improvements" of PER and CB over the case when there is no cache in the wireless terminal.

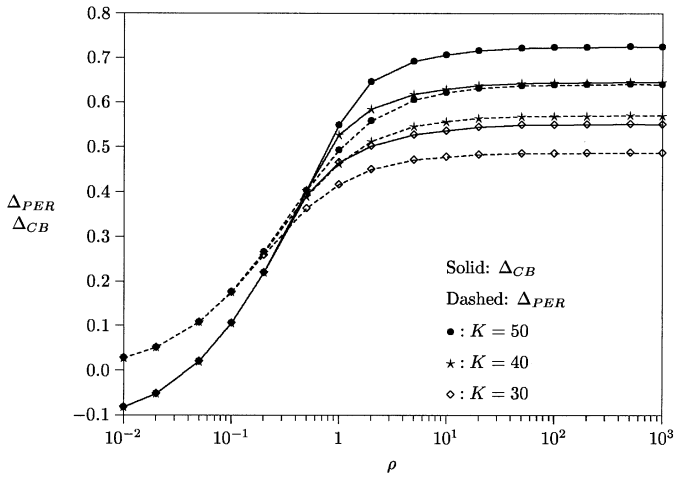

Fig. 13. Improvements of PER and CB over the cases when there is no cache ($N = 100, \alpha = 0.8, \nu = 1/\lambda^2$).

Let $C_{\rm NO}$ be the cost of accessing an object when there is no cache. Then

$$\Delta_{\rm PER} = \frac{C_{\rm NO} - C_{\rm PER}}{C_{\rm NO}} \quad \text{and} \quad \Delta_{\rm CB} = \frac{C_{\rm NO} - C_{\rm CB}}{C_{\rm NO}}.$$

Note that from (25), $C_{\rm NO} = 727 + 45 = 772$ B. Fig. 13 plots $\Delta_{\rm PER}$ and $\Delta_{\rm CB}$ against ρ. The figure indicates that PER is always better than the case when there is no cache. When $\rho < 0.1$, CB's performance is worse than the case when there is no cache. When $\rho > 10$, the improvements $\Delta_{\rm PER}$ and $\Delta_{\rm CB}$ can be in the range of 40%–70%. In this case, $C_{\rm PER} > C_{\rm CB}$. That is, 40%–70% of the wireless bandwidth can be saved by exercising the strongly consistent data access protocols with LRU cache replacement.

VII. Conclusion

This paper investigated the performance of wireless data access with cache. The LRU replacement policy is considered in our study. To support strongly consistent data access, two data access algorithms, PER and CB, were studied. The cache performance is measured by the effective cache hit ratio ($p_{\rm PER}$ and $p_{\rm CB}$) and the wireless transmission costs ($C_{\rm PER}$ and $C_{\rm CB}$). We used the business-card service to illustrate the performance of wireless data access. This application is a generalization of phone book, a popular feature of mobile terminals. Let ρ be the access-to-update ratio [see (24)]. Our study indicated the following.

- When ρ is small, the effective cache hit ratio is affected by the data updates and is insignificantly affected by the cache size. On the other hand, when ρ is large, the effective hit ratio is affected by the cache size, and is insignificantly affected by the data update frequency.
- When ρ is very large or very small, the effective hit ratios $p_{\rm PER}$ and $p_{\rm CB}$ have similar performance. When ρ has a moderate value (e.g., $0.2 < \rho < 100$), the data access protocols PER and CB have different effects on the effective hit ratios. In particular, $p_{\rm CB}$ is always larger than $p_{\rm PER}$.
- For the wireless transmission costs $C_{\rm CB}$ and $C_{\rm PER}$, except for a very large ρ, we observed that when the temporal locality α is small, $C_{\rm PER} < C_{\rm CB}$. When α is large

$C_{\mathrm{PER}} > C_{\mathrm{CB}}$. C_{PER} is always smaller than C_{NO} (the cost for the case when there is no cache). When ρ is very small, it is possible that $C_{\mathrm{CB}} > C_{\mathrm{NO}}$. When ρ is very large, significant improvements of PER and CB are observed over the case when there is no cache. In this case, $C_{\mathrm{PER}} > C_{\mathrm{CB}}$.

As we mentioned before, many wireless applications exhibit temporal locality, and the number N of the potential data objects that may be accessed by a wireless terminal is not significantly larger than the cache size. Our study indicated that if N is no more than twice of the cache size and the data access is reasonably frequent, then more than 50% of effective cache hit ratio can be observed. This result is consistent with the experiments reported in the WAP white paper [23].

ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers. Their comments have significantly improved the quality of this paper. Partial work of this paper was developed by K. H. Chang under the guidance of Y.-B. Lin.

REFERENCES

[1] L. Berslau, P. Cao, L. Fan, G. Phillips, and Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 1999, pp. 126–134.
[2] G. Cao, "A scalable low-latency cache invalidation strategy for mobile environments," in *ACM Mobicom*, Aug. 6–11, 2000, pp. 200–209.
[3] G. H. Cao, "Proactive power-aware cache management for mobile computing systems," *IEEE Trans. Comput.*, vol. 51, pp. 608–621, June 2002.
[4] K.-H. Chang, "Effects of cache size on data access for wireless application protocol (WAP)," M.S. thesis, Dept. Comput. Sci. and Inform. Eng., National Chiao Tung Univ., Hsinchu, Taiwan, R.O.C., 2001.
[5] F. Dawson, "vCard MIME Directory Profile," Internet Engineering Task Force, RFC 2426, 1998.
[6] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West, "Scale and performance in a distributed file system," *ACM Trans. Comput. Syst.*, vol. 6, no. 1, pp. 51–58, Feb. 1988.
[7] A. Kahol, S. Khurana, S. Gupta, and P. Srimani, "An efficient cache maintenance scheme for mobile environment," in *Proc. Int. Conf. Distributed Computing Systems*, Apr. 2000, pp. 530–537.
[8] F. P. Kelly, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.
[9] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[10] Y.-B. Lin, "Determining the user locations for personal communications networks," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 466–473, Aug. 1994.
[11] Y.-B. Lin and Y.-C. Chang, "Modeling frequently accessed wireless data with weak consistency," *J. Inform. Sci. Eng.*, vol. 18, pp. 581–600, 2002.
[12] Y.-B. Lin and I. Chlamtac, *Wireless and Mobile Network Architectures*. New York: Wiley, 2001.
[13] Y.-B. Lin, W. R. Lai, and R. J. Chen, "Performance analysis for dual band PCS networks," *IEEE Trans. Comput.*, vol. 49, pp. 148–159, Feb. 2000.
[14] M. Nelson, B. Welch, and J. Ousterhout, "Caching in the sprite network file system," *ACM Trans. Comput. Syst.*, vol. 6, no. 1, pp. 134–154, Feb. 1988.
[15] C. H. Rao, D.-F. Chang, and Y.-B. Lin, "iSMS: An integration platform for short message service and IP networks," *IEEE Network*, vol. 15, pp. 48–55, Mar./Apr. 2001.
[16] C. H. Rao, Y.-H. Cheng, K.-H. Chang, and Y.-B. Lin, "iMail: A WAP mail retrieving system," *Inform. Sci.*, vol. 151, pp. 71–91, 2002.
[17] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1996.
[18] M. Satyanarayanan *et al.*, "Coda: A highly available file system for a distributed workstation environment," *IEEE Trans. Comput.*, vol. 39, pp. 447–459, Apr. 1990.
[19] A. Silberschatz, J. L. Peterson, and P. Galvin, *Operating System Concepts*, 3rd ed. Reading, MA: Addison-Wesley, 1991.
[20] H. Stone, *High-Performance Computer Architecture*. Reading, MA: Addison-Wesley, 1990.
[21] (1998) Wireless Application Protocol Architecture Specification. WAP Forum Tech. Report. [Online]Available: http://www.wapforum.org
[22] (1998) Wireless Application Protocol Cache Model Specification. WAP Forum Tech. Report. [Online]Available: http://www.wapforum.org
[23] (1999) Wireless Application Protocol White Paper. WAP Forum Tech. Report. [Online]Available: http://www.wapforum.org
[24] (1999) Wireless Application Protocol V1.1 to V1.2. WAP Forum Tech. Report. [Online]Available: http://www.wapforum.org
[25] J. Yin, L. Alvisi, M. Dahlin, and C. Lin, "Volume leases for consistency in large-scale systems," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 563–576, July/Aug. 1999.
[26] G. K. Zipf, "Relative frequency as a determinant of phonetic change," *Reprinted from the Harvard Studies in Classical Philiology*, vol. XL, 1929.

Yi-Bing Lin (M'96–S'96–F'03) received the B.S.E.E. degree from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1983, and the Ph.D. degree in computer science from the University of Washington, Seattle, in 1990.

From 1990 to 1995, he was with the Applied Research Area at Bell Communications Research (Bellcore), Morristown, NJ. In 1995, he was appointed Professor in the Department of Computer Science and Information Engineering (CSIE), National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C. In 1996, he was appointed as Deputy Director of Microelectronics and Information Systems Research Center, NCTU. From 1997 to 1999, he was the Chairman of the CSIE, NCTU. His current research interests include design and analysis of personal communications services network, mobile computing, distributed simulation, and performance modeling. He has published over 150 journal articles and more than 200 conference papers.

Wei-Ru Lai received the B.E. degree and the Ph.D. degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1991 and 1999, respectively.

From 1999 to 2001, she was an Assistant Professor in the Department of Information Management, Chin-Min College, Miaoli, Taiwan, R.O.C. From 1999 to 2000, she was the Chairman of that department. She is currently an Assistant Professor in the Electrical Engineering Department, Yuan Ze University, Chun-Li, Taiwan, R.O.C. Her current research interests include design and analysis of personal communications service networks.

Jen-Jee Chen received the B.E. degree in computer science and information engineering from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 2001. He is currently working toward the Ph.D. degree at NCTU.

His research interests include design and analysis of a personal communications services network, computer telephony integration, mobile computing, and performance modeling.