

Location Tracking in PCS Networks Using Enhanced Caching

Po-Jen Chuang*, Shien-Da Chang and Tun-Hao Chao

*Department of Electrical Engineering, Tamkang University,
Tamsui, Taiwan 251, R.O.C.*

Abstract

In personal communication service (PCS) networks, the user location tracking strategy plays a critical role in ensuring successful mobile communication in any place at any time. The IS-41 scheme in North America and the MAP of the GSM system in Europe are two current standard location tracking schemes. To reduce communication cost for the two basic schemes, several location tracking strategies, either caching or pointer forwarding strategies, have been proposed. Caching strategies aim to reduce the cost occurring upon call delivery; pointer forwarding strategies intend to reduce the cost due to registration. This paper presents a new caching strategy which uses forwarding pointers, instead of updating users' location information, to locate users who make frequent moves while using PCS. Enhanced by forwarding pointers, the new caching strategy is able to reduce the cost resulting from registration, cache hit but invalid and cache miss. On call delivery, we update the cache records based on the fact that users tend to revisit previous RAs and to visit previously called RAs, and thus result in higher cache hit and valid ratios. A thorough performance analysis on different location tracking strategies is conducted to prove the favorable performance and reduced cost of the proposed caching strategy.

Key Words: Personal Communication Service (PCS) Networks, Location Tracking Strategies, Home Location Registers (HLRs), Visitor Location Registers (VLRs), Call Delivery, Registration, Performance Analysis, Call-to-mobility Ratios, Cache Hit and Miss Ratios, Location Tracking Cost

1. Introduction

In a Personal Communication Service (PCS) network [1], the home location register (HLR) resides in the service control point (SCP) which connects to a regional signal transfer point (RSTP) as depicted in Figure 1. Through a connection network, the RSTP connects to all local STPs (LSTP) in the region, performing message routing translation and screening functions in the Signaling System 7 (SS7) network. The individual STPs actually represent the mated-pair configuration. Every LSTP comprises an LATA geographical area which connects

to multiple service switching points (SSPs). The mobile switching center (MSC) and a visitor location register (VLR) are collocated with an SSP. A registration area (RA) consists of one or more radio port coverage areas or cells [2]. We assume that each RA is served by a single VLR. The MSC of an RA is responsible for maintaining and accessing the VLR and for switching between radio ports. The VLR associated with an RA is responsible for maintaining a subset of the user information contained in the HLR [3]. The VLR is collocated with the MSC and the MSC/VLR combination will evolve to be SS7 compatible.

How to track a user's location in the PCS network may significantly affect the quality of mobile communi-

*Corresponding author. E-mail: pjchuang@ee.tku.edu.tw

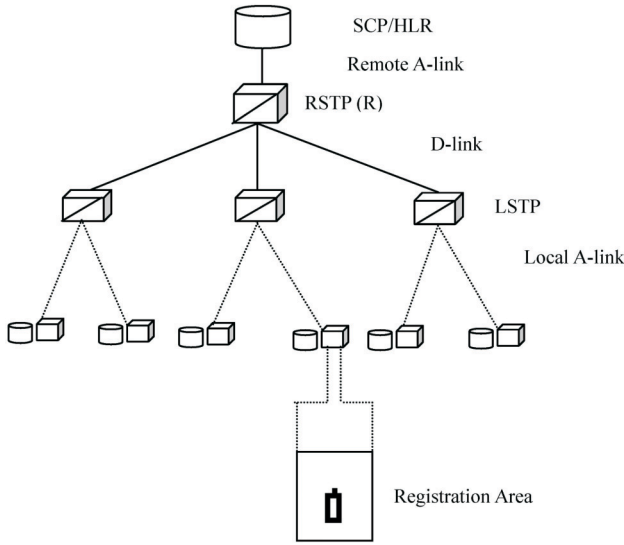


Figure 1. The PCS network architecture.

cation. The established standard location tracking schemes include the IS-41 scheme [4–6] in North America and the MAP of the GSM system [7,8] in Europe. To augment the two basic schemes, i.e., to reduce the network communication cost, several user location tracking strategies -- categorized as caching strategies and pointer forwarding strategies (e.g. [9–20]) -- have been introduced. Caching strategies aim to reduce the cost due to call delivery, while pointer forwarding strategies attempt to reduce the cost of registration.

A caching strategy suits users who receive calls more frequently (i.e., with high users' call-to-mobility ratios (CMRs)) than change the PCS registration areas. For users with CMRs around 5 and stable calls and moving patterns, caching brings about 20–60% cost reduction over the basic location tracking scheme [1]. Reducing registration (i.e., cutting down HLR updates) will definitely improve performance for caching strategies.

Pointer forwarding strategies, by contrast, are suitable for users who move among the PCS registration areas frequently but receive calls not as frequently (i.e., with low CMRs). For users with CMRs less than 1, a pointer forwarding strategy saves cost over the basic strategy without increasing the mean call setup time [15].

Based on the above observation, we propose a new location tracking strategy to locate users who move from place to place while using PCS. The proposed strategy, a

caching strategy enhanced by forwarding pointers, tends to deploy the cache device at switch MSC/VLR on a system-wide basis. The cached records are kept at MSC/VLR from which calls originate and which may or may not be the switch where the call receiver is currently registered or resides [1]. In many cases, it is indeed more feasible to save and reuse a user's location information obtained during his previous incoming call than to query his HLR. When *cache hit but invalid* happens (after a user moves to another RA), we use forwarding pointers which are recorded in cache, instead of updating the user's location information to the HLR, to locate the user and as a result efficiently reduce the cost due to registration.

It has been observed that population movements tend to gather in certain areas, such as residences, workplaces and shopping centers, where people stay for a considerable period of time [14]. (RAs containing these areas are called attraction sites.) It is also observed that people may visit RAs which they previously called. In the cache part of our strategy, we make use of these two features -- that users tend to revisit previous RAs and to visit previously called RAs -- to update the cached records and result in higher *cache hit and valid* ratios. Besides, the forwarding pointers in our strategy also help reduce the penalty due to cache hits but invalid, further cutting down the communication cost.

A thorough analytic evaluation is conducted to compare the performance of our strategy and other location tracking strategies. The results show that compared with other strategies, our proposed strategy yields favorable performance (higher location hit probability) at reduced registration, traversal and total costs.

This paper is organized as follows. Section 2 presents the proposed location tracking strategy. Section 3 gives the analytic models. A thorough performance analysis and comparison between related strategies is provided in Section 4. Section 5 concludes the paper.

2. The Proposed Location Tracking Strategy

The proposed strategy is built aiming to reduce the penalty of invalid cache records occurring in previous caching strategies due to users' movements. It can fur-

ther avoid message traffic bottlenecks and cut down database accesses for querying the HLR to locate a called subscriber. Besides using the effective information caching to avoid unnecessary message traffic, our new strategy meanwhile cuts out the operation of updating a user's HLR (when he moves across RAs) by using forwarding pointers recorded in cache entries. Two operations incurring signaling have been defined in [1] and [15]:

1. **MOVE** – (for the PCS user) to move from one RA to another;
2. **FIND** – to determine the RA in which the PCS user is currently located.

Our new strategy, like previous caching strategies, saves a user's location information obtained during his previous incoming call in a cache entry and reuse it without querying his HLR. Such cached information is useful only for users who receive calls more frequently than change registration areas. For users who receive calls infrequently but change registration areas frequently, our strategy works as well because, unlike previous caching strategies, it records forwarding pointers in cache entries to locate the users.

In our strategy, the entry of the cache in the VLR is defined as

UID	NEXT_VLR
-----	----------

UID records the user's mobile identification number (MIN). NEXT_VLR records the user's current residing VLR identifier (recorded at the user's previous call) and may represent a forwarding pointer pointing to the next VLR in the pointer chain (recorded at the user's each movement).

When a call is to be delivered, the procedure to locate the called party in our strategy is defined in Figure 2.

When a user moves from one RA to another, the operating steps in our new strategy is thus defined in Figure 3.

The MOVE and FIND procedures in our strategy and in pointer forwarding strategies [11,15] are compared in Figures 4 and 5.

As Figure 4 indicates, the MOVE procedure of our

strategy (depicted by the big bold cross sign) sets no threshold on the allowed forwarding pointers (because the cache record update helps limit the length of the pointer chain, to be discussed later). Without such a threshold, we are able to reduce unnecessary updates on users' location information in the HLR.

Figure 5 shows that the HLR in a pointer forwarding strategy needs to be queried upon any incoming call to get the recorded forwarding pointer (steps (1) and (2)) and then to be updated accordingly (steps (3) and (4)). In total, a pointer forwarding strategy will take 5 steps ((1), (2), (2'), (3) and (4)) to find out the RA where the user is currently located. Our new strategy, by contrast, takes only step (2') to locate the called user (when not in the rare cache miss situation) as it records (a) the called user's location (obtained during the previous call) in the caller's switch, and (b) forwarding pointers to the called user's switch as NEXT_VLRs in the cache entries of switches along which the called user moves. The location tracking cost due to HLR queries and updates (indicated by the bold cross signs) is thus saved.

3. The Analytic Models

3.1 Analytic Models for Caching Structures

The T-threshold caching strategy in [2] derives probability α for cache hit with threshold T, probability β for invalid cache records and probability γ for cache miss as follows.

$$\alpha = \Pr[t_C < t_m \text{ and } t_C \leq T] = \frac{\lambda_C}{\lambda_C + \lambda_m} [1 - e^{-(\lambda_C + \lambda_m)T}] \quad (1)$$

$$\beta = \Pr[t_C \geq t_m \text{ and } t_C \leq T] = 1 - e^{-\lambda_C T} - \frac{\lambda_C}{\lambda_C + \lambda_m} [1 - e^{-(\lambda_C + \lambda_m)T}] \quad (2)$$

$$\gamma = \Pr[t_C > T] = e^{-\lambda_C T} \quad (3)$$

To suit our need, we redefine the three probabilities:

α : the probability for cache hit with valid cache records

β : the probability for cache hit with invalid cache records

```

FIND(){
  A call to a PCS user is detected at the local switch;
  if (the called user is in the same RA) return;
  The switch searches the entry of the called user in the cache;
  if (there's a cache entry for the called user){ /* cache hit */
    The switch queries the VLR specified in the cache entry;
    if (the current VLR of the called user is at the queried VLR){
      /* cache hit and valid */
      The queried VLR returns the called user's location to the calling switch;
      The called user's VLR updates the cache entry of the calling user;
    }
    else { /* cache hit but invalid */
      do { /* apply pointer forwarding to find the current VLR */
        the queried VLR queries the next VLR in the pointer chain;
      } while (the called user is not at the queried VLR);
      /* the current VLR of the called user is found */
      The current VLR of the called user returns the called user's location to the calling switch;
      The calling user's VLR updates the cache entry of the called user;
      The called user's VLR updates the cache entry of the calling user;
    }
  }
  else { /* cache miss */
    The switch queries the called user's HLR;
    The HLR queries the VLR specified in the HLR;
    if (the current VLR of the called user is the queried VLR){
      /* HLR up-to-date */
      The HLR returns the called user's location to the calling switch;
      The calling user's VLR updates the cache entry of the called user;
      The called user's VLR updates the cache entry of the calling user;
    }
    else { /* HLR out-of-date */
      do { /* apply pointer forwarding to find the current VLR */
        the queried VLR queries the next VLR in the pointer chain;
      } while (the called user is not at the queried VLR);
      /* the current VLR of the called user is found */
      The current VLR of the called user returns the called user's location to the calling switch;
      The called user's HLR updates the location of the called user;
      The calling user's VLR updates the cache entry of the called user;
      The called user's VLR updates the cache entry of the calling user;
    }
  }
}

```

Figure 2. The FIND() procedure of our strategy.

```

MOVE() {
  The user registers at the new VLR and passes the id of the former VLR to it;
  The new VLR deregisters the user at the old VLR;
  The old VLR updates the cache entry of the moving user and sends ACK
  and the user's service profile to the new VLR;
}

```

Figure 3. The MOVE() procedure of our strategy.

γ : the probability for cache miss (i.e., no cache records)

The T-threshold in (1), (2) and (3) is chosen from t_c and will be compared with the duration of cache records. t_c re-

presents the time interval between two consecutive calls directed from an RA to a remote user. To derive a general caching formula for our strategy, we adopt the same analytic model using t_c but leave out the T-threshold.

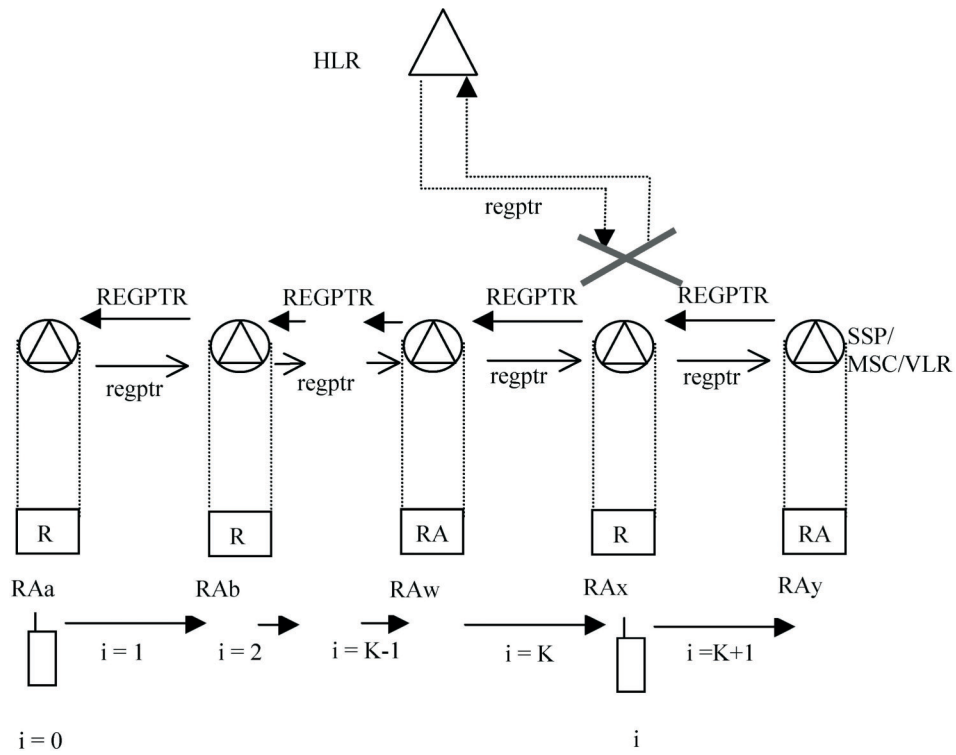


Figure 4. The MOVE() procedure comparison between our strategy and the pointer forwarding strategies.

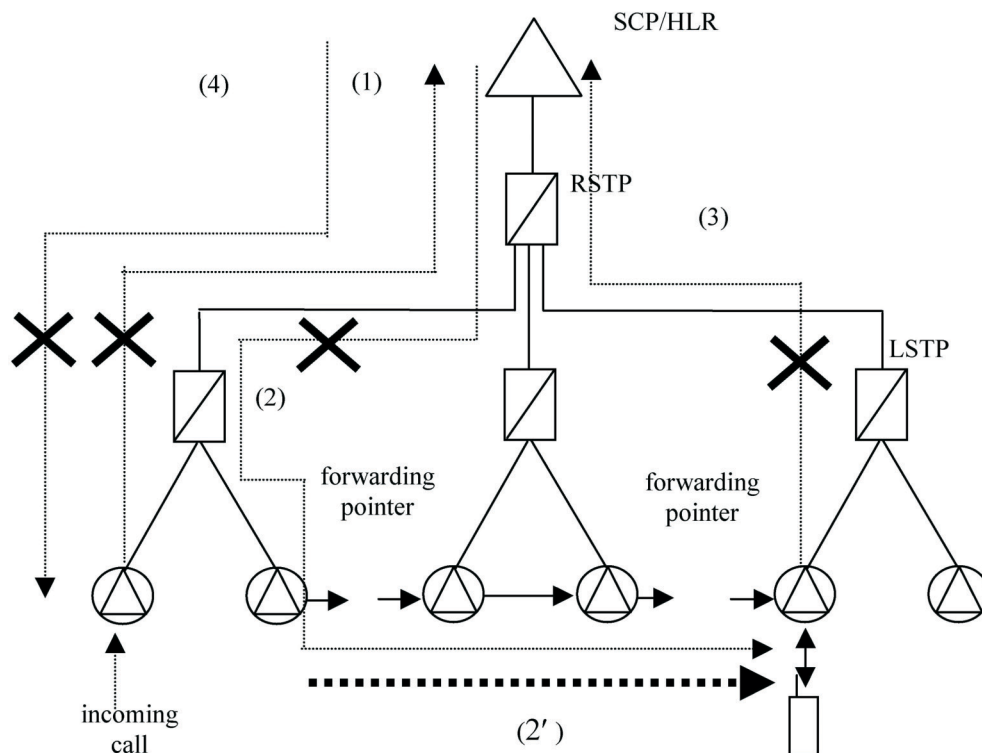


Figure 5. The FIND() procedure comparison between our strategy and the pointer forwarding strategies.

In our analytic model,

t_c is an independent and identically distributed random variable, representing the time interval between two consecutive calls directed from an RA to a remote user,

t_m is an independent and identically distributed random variable, representing the time interval when the user resides in an RA, and

t_m is the time interval between the previous call to the user and the user's moving out of the RA.

$f_c(t)$, $f_M(t)$, and $f_m(t)$ are respectively the density functions of t_c , t_M and t_m . If the incoming calls are random observers (for example, forming a Poisson process) of t_M [2], then

$$f_m(t) = \frac{1}{E[t_M]} \int_{t_1=t}^{\infty} f_M(t_1) dt_1 \quad (4)$$

Let $f_h(t_c)$ be the density function of t_c when $t_c < t_m$. Then,

$$f_h(t_c) = f_c(t_c) \int_{t_m=t_c}^{\infty} f_m(t_m) dt_m$$

Assume that when a user moves out of a certain RA, it will not return to it again during the time period. (The effect that a user may return to a previous RA can be easily modeled by a two-dimensional random walk [15].) Under such an assumption, the derived hit ratio will be slightly lower than the true one. **Probability** α for cache hit with valid cache record can be thus derived:

$$\begin{aligned} \alpha &= \Pr[t_c < t_m] = \int_{t_c=0}^{\infty} f_h(t_c) dt_c \\ &= \int_{t_c=0}^{\infty} [f_c(t_c) \int_{t_m=t_c}^{\infty} f_m(t_m) dt_m] dt_c \end{aligned} \quad (5)$$

Assume that t_c is exponentially distributed with mean $1/\lambda_c$, t_M is exponentially distributed with mean $1/\lambda_m$, and t_m is also exponentially distributed. Because $f_m(t) = f_M(t)$, (5) can be rewritten as

$$\begin{aligned} \alpha &= \int_{t_c=0}^{\infty} \lambda_c e^{-\lambda_c t_c} \left[\int_{t_m=t_c}^{\infty} \lambda_m e^{-\lambda_m t_m} dt_m \right] dt_c \\ &= \int_{t_c=0}^{\infty} \lambda_c e^{-\lambda_c t_c} \lambda_m e^{-\lambda_m t_c} dt_c = \frac{\lambda_c}{\lambda_c + \lambda_m} [1 - e^{-(\lambda_c + \lambda_m)t_c}] \end{aligned} \quad (6)$$

Likewise, **probability** β for cache hit with invalid cache records will become

$$\begin{aligned} \beta &= \Pr[t_c \geq t_m] = \int_{t_c=0}^{\infty} [f_c(t_c) - f_h(t_c)] dt_c \\ &= 1 - e^{-\lambda_c t_c} - \frac{\lambda_c}{\lambda_c + \lambda_m} [1 - e^{-(\lambda_c + \lambda_m)t_c}] \end{aligned} \quad (7)$$

As mentioned, invalid cache records will result in “location miss” and extra cost penalty (such as HLR queries) for caching strategies. For improvement, our caching strategy employs enhanced pointers (NEXT_VLRs in cache entries) to locate the user (to achieve “location hit” in this situation) and hence to reduce the penalty of location miss in other caching strategies.

Probability γ for no cache records (cache miss), given below, indicates the call from some caller to the called user has not yet arrived and the cache record in the called user's switch for the caller is still missing.

$$\begin{aligned} \gamma &= \Pr[t_c < t_m \text{ and } t_m - t_m = t_m \text{ (i.e., } t_m = 0)] \\ &= \int_{t_c=0}^{\infty} f_c(t_c) dt_c = e^{-\lambda_c t_c} \end{aligned} \quad (8)$$

3.2 Cache Record Updates in Our Strategy

To illustrate cache record updates in our strategy, an example case -- a user's portable movement sequence between two phone calls -- is given in Figure 6. The figure shows an 8×8 restriction area in which each RA sets up a VLR to maintain the user's location information. The user moves from R_0 to R_8 by way of R_6 (which keeps his cache record), and the cache record will be updated when he moves to R_7 . In this case, we are interested to know

- (1) which RAs have the user's cache records before he moves out of R_0 , and
- (2) the user's movement route after he moves out of R_0 between two phone calls.

We then define

$$\alpha_{K+1} = m_{K+1} \times n \quad (9)$$

Where

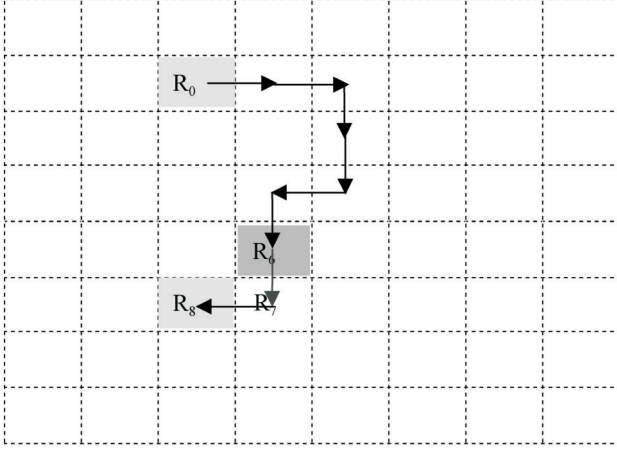


Figure 6. A user's cache record update during its movement (from R_0 to R_8) between two phone calls.

α_{K+1} = Prob [the user's cache record update]

m_{K+1} = Prob [the user's moving out of R_0 between two phone calls]

n = Prob [which RAs have the user's cache record]

The formulas for cache record updates can be derived from [16] and Figure 6. First, assume the incoming calls to a user are a Poisson process and the time he resides in an RA is generally distributed. Let t_c be the time interval between two consecutive phone calls to the user. Suppose the user resides in R_0 when making the previous call, visits another K RAs and then resides in the j th RA for a period t_{Mj} ($0 \leq j \leq K$) after the phone call.

Let

t_m be the time interval between the arrival of the previous call to a user and the moment he moves out of R_0 ;

$t_{c,i}$ be the time interval between the user's entering R_{Mi} and the arrival of the next phone call;

t_{Mi} be independently identically distributed random variables with general distribution $F_m(t_{Mi})$, density function $f_m(t_{Mi})$ and Laplace-Stieltjes Transform.

$$f_m^*(s) = \int_{t=0}^{\infty} e^{-st} f_m(t) dt$$

Let

$f_c(t)$ and $r_m(t)$ be the density functions of t_c and t_m respectively, and get

$$E[t_c] = 1/\lambda_c \text{ and } E[t_{Mi}] = 1/\lambda_m.$$

Since the incoming phone calls are a Poisson process, we have $f_c(t) = \lambda_c e^{-\lambda_c t}$.

Based on the memoryless property of the exponential distribution, $t_{c,i}$ has the same exponential distribution as t_c for all i 's, and based on the random observer property in [16],

$$r_m(t) = \lambda_m \int_{\tau=t}^{\infty} f_m(\tau) d\tau = \lambda_m [1 - F_m(t)] \quad (10)$$

The Laplace-Stieltjes Transform for the t_m distribution will be

$$\begin{aligned} r_m^*(s) &= \int_{t=0}^{\infty} e^{-st} r_m(t) dt \\ &= \int_{t=0}^{\infty} e^{-st} \lambda_m [1 - F_m(t)] dt \\ &= \frac{\lambda_m}{s} - \int_{t=0}^{\infty} e^{-st} \lambda_m F_m(t) dt \\ &= \frac{\lambda_m}{s} - \left[\frac{\lambda_m}{s} e^{-st} F_m(t) \Big|_{t=0}^{\infty} - \frac{\lambda_m}{s} \int_{t=0}^{\infty} e^{-st} f_m(t) dt \right] \\ &= \frac{\lambda_m}{s} \left[1 - \int_{t=0}^{\infty} e^{-st} f_m(t) dt \right] \\ &= \frac{\lambda_m}{s} [1 - f_m^*(s)] \end{aligned} \quad (11)$$

If a user is at the K th movement when passing the RA with his cache record, focus on his $(K+1)$ th movement (to go out of the RA with the cache record) and get

$$\begin{aligned} m_{K+1} &= \Pr[t_{M1} + t_{M2} + \dots + t_{MK} < t_c \leq t_{M1} + t_{M2} + \dots + t_{MK+1}] \\ &= \left(\prod_{i=1}^{K+1} \Pr[t_{c,i} > t_{Mi}] \right) \Pr[t_{c,K} \leq t_{MK}] \\ &= (1 - f_m^*(\lambda_c)) (f_m^*(\lambda_c))^{K+1} \end{aligned} \quad (12)$$

$\Pr[t_{c,i} > t_{Mi}]$ and $\Pr[t_{c,k} \leq t_{MK}]$ are derived as follows.

$$\begin{aligned} \Pr[t_{c,i} > t_{Mi}] &= \int_{t_{M,i}=0}^{\infty} \int_{t_{c,i}=t_{M,i}}^{\infty} \lambda_c e^{-\lambda_c t_{c,i}} f_m(t_{M,i}) dt_{M,i} dt_{c,i} \\ &= f_m^*(\lambda_c) \end{aligned} \quad (13)$$

$$\Pr[t_{c,K} \leq t_{MK}] = 1 - \Pr[t_{c,K} > t_{MK}] = 1 - f_m^*(\lambda_c) \quad (14)$$

Assume the user's residence times are with Gamma distribution. (Most studies use exponential distribution, a special case of Gamma, for its simplicity.) Here, with

mean $1/\lambda_m$ and variance V , the Laplace-Stieltjes Transform of a Gamma random variable is expressed as

$$f_m^*(s) = \left(\frac{\lambda_m \gamma}{s + \lambda_m \gamma} \right)^\gamma, \text{ Where } \gamma = \frac{1}{V\lambda_m^2}.$$

We know that the user's movement between two phone calls is significantly affected by the variance of residence times. When $s = \lambda_c$, $V = \frac{1}{\lambda_m^2}$ and $\gamma = 1$, we have

$$f_m^*(\lambda_c) = \left(\frac{\lambda_m}{\lambda_c + \lambda_m} \right)$$

m_{K+1} of (9) can now be derived as

$$m_{K+1} = \left(1 - \frac{\lambda_m}{\lambda_c + \lambda_m} \right) \left(\frac{\lambda_m}{\lambda_c + \lambda_m} \right)^{K+1} \quad (15)$$

The probability for the user's cache record update in this example case ($n = 1/64$) is then obtained:

$$\alpha_{K+1} = m_{K+1} \times n = \left(1 - \frac{\lambda_m}{\lambda_c + \lambda_m} \right) \left(\frac{\lambda_m}{\lambda_c + \lambda_m} \right)^{6+1} \times \frac{1}{64}$$

We can thus conclude: The probability of which RAs having the user's cache record depends on the number of VLRs in the PCS network, and so does the probability of a user's cache record updates.

3.3 Performance Analysis on Forwarding Pointer Structures

In cache miss cases

To analyze the performance of forwarding pointers under *cache miss* in our enhanced caching strategy, we first characterize users by their call-to-mobility ratios (CMRs). The CMR [10] is defined as the expected number of calls to a user during his stay at a given RA. (Note that the CMR is defined here as the calls received by a particular user, not the calls originating from the user.) If the user's call arrival rate is a mean rate λ , or λ_c , and the time the user resides in a given RA has a mean $1/\mu$, or $1/\lambda_m$, then the CMR, denoted as p , will be $CMR = p = \lambda/\mu$.

To conduct cost comparison between our caching strategy and the IS-41 scheme, we need to model the procedure of IS-41. Recall that when a user crosses several RAs between two consecutive calls, IS-41 needs to update the user's HLR each time he moves to a new RA. By contrast, our strategy is able to avoid unnecessary HLR updates as it sets up forwarding pointers in all moves to locate the user.

Let C_B and $C_{C,F}$ respectively be the total cost for IS-41 and for our enhanced caching strategy in maintaining user information (location update) and locating the user (location tracking) between two consecutive calls. A number of notations are listed below to facilitate later discussion.

- $M_{C,F}$:** the expected cost of all enhanced FwdMOVE between two consecutive calls in our strategy
- $F_{C,F}$:** the average cost of the enhanced FwdFIND in our strategy
- M :** the total cost of all BasicMOVEs between two consecutive calls
- m :** the cost of a single invocation of BasicMOVE
- F :** the cost of a single BasicFIND
- S :** the cost of setting up a forwarding pointer between VLRs during a FwdMOVE
- T :** the cost of traversing a forwarding pointer between VLRs during a FwdFIND
- K :** the number of RAs a user moves across between two consecutive calls; also the number of forwarding pointers

Thus we have

$$C_B = M + F = m/p + F \quad (16)$$

$$C_{C,F} = M_{C,F}/p + F_{C,F} \quad (17)$$

Now let $\alpha(K)$ be the probability that there are K RA crossings between two call arrivals, expressed in [10, 15] as

$$\alpha(i) = \frac{\mu}{\lambda} [1 - fm^*(\lambda)]^2 [fm^*(\lambda)]^{i-1} \quad (18)$$

When a user crosses K RA boundaries between two

consecutive calls, there will be K pointer creations in our strategy to avoid unnecessary HLR updates. Thus,

$$M_{C,F} = \sum_{i=0}^{\infty} iS\alpha(i) \quad (19)$$

Let $\Theta(K)$ be the number of pointers traced in the enhanced FwdFIND operation. Note that the last Basic-MOVE operation (if any), the user will have traversed K RA boundaries. But the enhanced FwdFIND operation will trace less than K RAs to locate the user if he revisits previously visited RAs during the K moves (i.e., 'loop' exists among the K moves). Even if the user never revisits previously visited RAs, $\Theta(K) = K$, and

$$F_{C,F} = \sum_{i=0}^{\infty} \Theta(K) T\alpha(i) + F \quad (20)$$

For better illustration, we assume that after the RA residence time of a user is Gamma distributed with mean $1/\mu$. (Gamma distribution is flexible in setting various parameters and can be used to fit the first two moments of the field data.) The Laplace transform of a Gamma distribution is $fm^*(s) = \left(\frac{\gamma\mu}{\lambda + \gamma\mu}\right)^\gamma$, thus we have

$$g = fm^*(\lambda) = \left(\frac{\gamma\mu}{\lambda + \gamma\mu}\right)^\gamma = \left(\frac{\gamma}{p + \gamma}\right)^\gamma$$

By setting $\gamma = 1$ (indicating an exponential distribution for the RA residence time), we have $g = \frac{1}{1+p}$ and may rewrite (19) and (20) as

$$M_{C,F} = \frac{S}{p} \quad (21)$$

$$F_{C,F} = \frac{T}{p} - \frac{(KT)}{[(1+p)^K - 1]} + F \quad (22)$$

$$C_{C,F} = \frac{S+T}{p} - \frac{(KT)}{[(1+p)^K - 1]} + F \quad (23)$$

In the basic IS-41 scheme, updating the HLR and

performing a BasicFIND involve the same number of messages between HLR and VLR databases, i.e., $m = F$. Without loss of generality, we normalize $m = 1$. We assume the cost of setting up a forwarding pointer is about twice the cost of traversing it (because twice as many messages are involved) and set $S = 2T$. We also let $S = \delta$ ($\delta < 1$) as registering at the HLR is more expensive than setting up forwarding pointers. From (16), (21), (22) and (23), we obtain

$$\frac{M_{C,F}}{M} = \delta \quad (24)$$

$$\frac{F_{C,F}}{F} = 1 + \frac{\delta}{2p} - \frac{\delta K}{2[(1+p)^K - 1]} \quad (25)$$

$$\frac{C_{C,F}}{C_B} = \frac{p}{1+p} \left\{ 1 + \frac{3\delta}{2p} - \frac{\delta K}{2[(1+p)^K - 1]} \right\} \quad (26)$$

In cache hit cases

For further comparison, performance formulas for pointer forwarding strategies under the cache hit condition are also derived. Suppose that the PCS network is a limited VLR architecture system with sufficient cache size devices. After a period of operation, users get their cache records and use the records to make phone calls. For example, probability γ for users with no cache records will equal to zero when $CMR > 1$. In this situation, users making a phone call need not query the HLR because they can use the cache records to locate the call receiver. Thus we obtain the cost formula for pointer traversal rates

$$M_{C,F} = \frac{S}{p} \quad (27)$$

$$F_{C,F} = \frac{T}{p} - \frac{(KT)}{[(1+p)^K - 1]} \quad (28)$$

$$C_{C,F} = \frac{S+T}{p} - \frac{(KT)}{[(1+p)^K - 1]} \quad (29)$$

Now we can rewrite (24), (25) and (26) into

$$\frac{M_{C,F}}{M} = \delta \quad (30)$$

$$\frac{F_{C,F}}{F} = \frac{\delta}{2p} - \frac{\delta K}{2[(1+p)^K - 1]} \quad (31)$$

$$\frac{C_{C,F}}{C_B} = \frac{p}{1+p} \left\{ \frac{3\delta}{2p} - \frac{\delta K}{2[(1+p)^K - 1]} \right\} \quad (32)$$

For easier reference, the definitions of important notations appearing in this section are put together and listed in Table 1.

4. Performance Evaluation

4.1 On Caching Strategies

For caching strategies, invalid cache records will result in location miss and extra cost penalty. To improve it, our strategy uses enhanced pointers (NEXT_VLRs in cache entries) to locate the user, and as a result reduces

the penalty of location miss.

The location hit probability

The probability of location hit for our enhanced caching strategy includes both probability α (cache hit with valid cache records) and probability β (cache hit with invalid cache records), in contrast to probability α alone for the other caching strategies. Figures 7 and 8 give the location hit probabilities under $\text{CMR} = 2 \sim 10$ and $\text{CMR} = 0.1 \sim 1$ for the caching strategy in [3] (plotted as Old) and for our strategy (plotted as New). As the result shows, our strategy yields about 20% higher location hit probability when $\text{CMR} = 2 \sim 10$ due to the added enhanced pointers in cache entries. For $\text{CMR} \leq 1$, the location hit probability for our strategy increases about 50~70%. Such a remarkable lift is achieved partly because low CMR brings about low cache record valid probability in general caching strategies (due to the user's frequent change of PCS registration areas) and partly because enhanced pointers are employed. When location

Table 1. Definitions of important notations in Sec. 3

Notation	Definition
α	the probability for cache hit with valid cache records
β	the probability for cache hit with invalid cache records
γ	the probability for cache miss (i.e., no cache records)
λ_c	the user's call arrival rate
λ_m	the user's mobility rate, i.e., the time the user resides in a given RA has a mean $1/\lambda_m$
t_c	an independent and identically distributed random variable, representing the time interval between two consecutive calls directed from an RA to a remote user
t_M	an independent and identically distributed random variable, representing the time interval when the user resides in an RA
t_m	the time interval between the previous call to the user and the user's moving out of the RA.
$f_c(t)$	the density functions of t_c
$f_M(t)$	the density functions of t_M
$f_m(t)$	the density functions of t_m
CMR	the call to mobility ratio $= \lambda_c/\lambda_m$
C_B	the total cost for IS-41
$C_{C,F}$	the total cost for our strategy
$M_{C,F}$	the expected cost of all enhanced FwdMOVE between two consecutive calls in our strategy
$F_{C,F}$	the average cost of the enhanced FwdFIND in our strategy
M	the total cost of all BasicMOVES between two consecutive calls
m	the cost of a single invocation of BasicMOVE
F	the cost of a single BasicFIND
S	the cost of setting up a forwarding pointer between VLRs during a FwdMOVE
T	the cost of traversing a forwarding pointer between VLRs during a FwdFIND
K	the number of RAs a user moves across between two consecutive calls; also the number of forwarding pointers

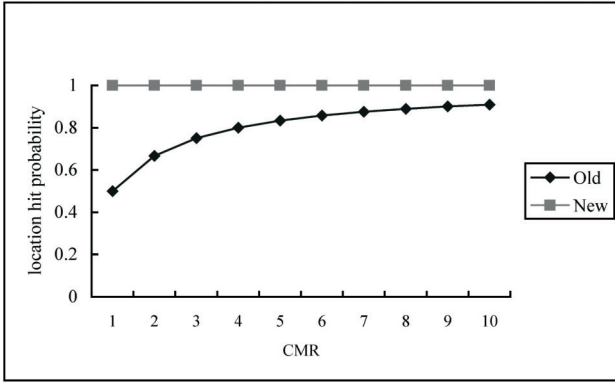


Figure 7. The location hit probabilities under CMR = 2~10 for our strategy (New) and the caching strategy in [3] (Old).

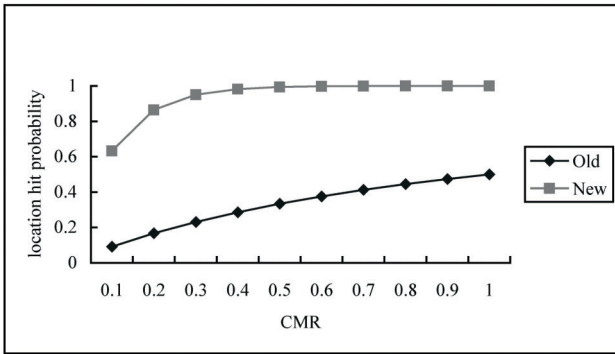


Figure 8. The location hit probabilities under CMR = 0.1~1 for our strategy (New) and the caching strategy in [3] (Old).

hit probabilities are significantly lifted, the penalty due to invalid cache records will also be reduced.

The cache miss probability

Figure 9 presents the probability of cache miss (no cache records), γ , under CMR = 0.1~1. (Note that due to unlimited cache sizes, γ will be zero when CMR > 1.)

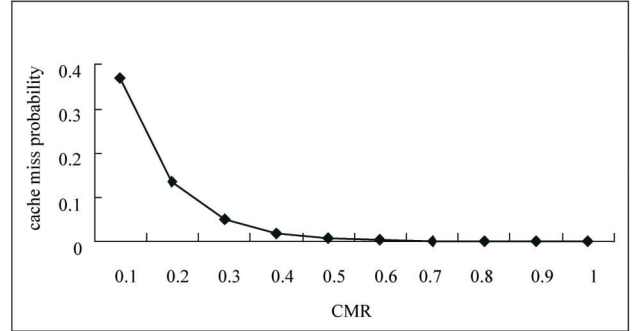


Figure 9. The cache miss probabilities under CMR = 0.1~1 for our strategy.

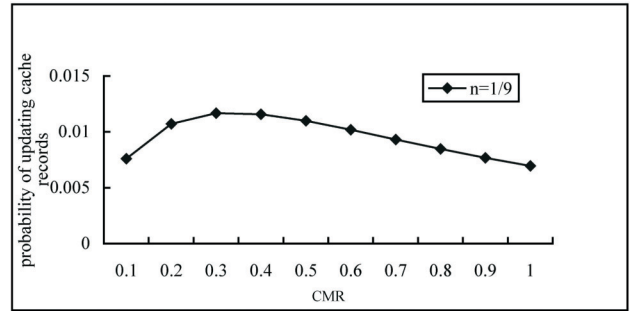


Figure 10. The mean probability of updating cache records under CMR = 0.1~1 for our strategy.

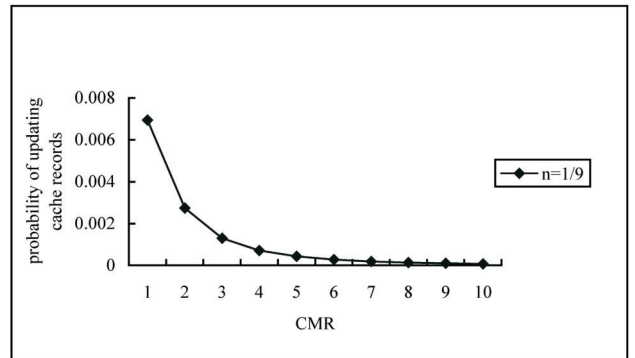


Figure 11. The mean probability of updating cache records under CMR = 1~10 for our strategy.

The probability of updating cache records

Figures 10 and 11 give the mean probabilities of updating cache records in a network with 9 VLRs under CMR = 0.1~1 and CMR = 1.5~10. As mentioned, our caching strategy updates cache records based on the fact that users will visit previously called RAs and thus results in higher probability for cache hit with valid cache records, α . It is also specified that the probability of which RA's having a user's cache record depends on the

number of VLRs in the PCS network. Now probability α_{K+1} is evaluated under probability $n = 1/9$. The result in Figure 10 shows that users who change PCS registration areas frequently but receive calls less frequently (with low CMRs) have higher probability α_{K+1} . Figure 11 gives the contrasting case: Users who change PCS registration areas infrequently but receive calls frequently (with high CMRs) get lower probability α_{K+1} . This is because the results, which depend on the users' moving

patterns, are decreasing functions of CMRs. The probabilities for a user to update his cache record in networks with different numbers of VLRs (9, 25, 49, 81, etc.) are collected in Figures 12 and 13 under $\text{CMR} = 0.1 \sim 1$ and $\text{CMR} = 1 \sim 10$. The result shows that the more VLRs a network has, the less likely a user updates his cache record, and vice versa.

4.2 On Pointer Forwarding Structures

Performance comparison between our strategy and several pointer forwarding structures is conducted under 3 situations: *Cache hit with valid cache records*, *cache hit with invalid cache records* and *cache miss*. The performance parameters of interest are the tracking cost and the total cost. Cost comparisons are conducted under a specific model based on the architecture in Figure 1.

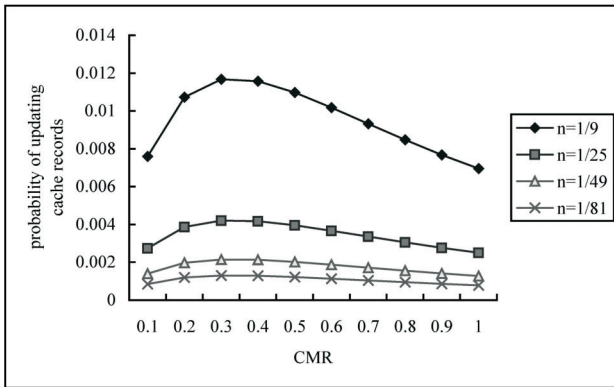


Figure 12. The probability of updating cache records under $\text{CMR} = 0.1 \sim 1$ and different VLR architectures for our strategy.

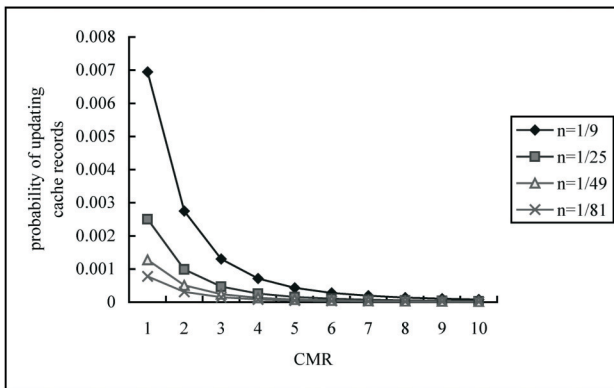


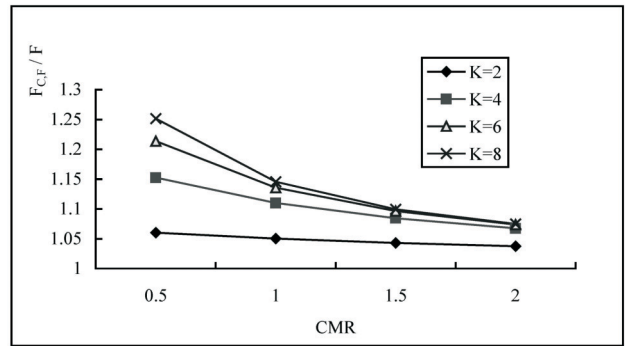
Figure 13. The probability of updating cache records under $\text{CMR} = 1 \sim 10$ and different VLR architectures for our strategy.

In cache miss cases

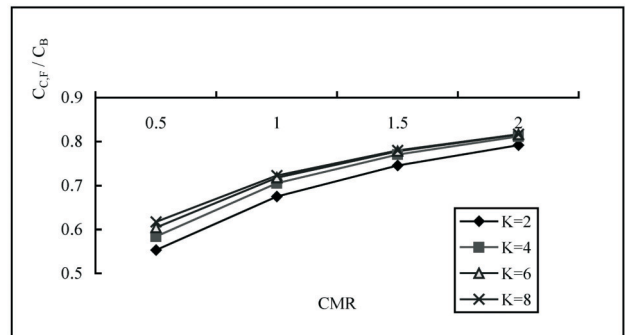
The following are some background assumptions mentioned in Section 3:

- (a) Because updating the HLR and performing a BasicFIND involve the same number of messages between HLR and VLR databases, we set $m = F$.
- (b) Assuming that the cost for setting up a forwarding pointer is about twice the cost of traversing it (since twice as many messages are involved), we set $S = 2T$.
- (c) Normalize $m = 1$ and $S = \delta$, with $\delta < 1$.

Compared with the update cost of registering at an HLR in the IS-41 scheme, our enhanced caching strategy involves only the cost of setting up a forwarding pointer -- δ , which is less than 1. Without having to register at each HLR helps us reduce the update cost significantly. The tracking cost and the total cost for our strategy and for the IS-41 scheme obtained under $\text{CMR} \leq 2$ are provided and compared in Figure 14. In Figure 14 (a), the



(a)



(b)

Figure 14. The FIND and total costs at *cache miss* for our strategy and the IS-41 scheme with $\delta = 0.3$ and $\text{CMR} \leq 2$. (a) the FIND cost F_{CF}/F . (b) the total cost C_{CF}/C_B .

tracking cost for our strategy appears higher than that for IS-41. This is because a call to the user in our strategy needs to traverse the pointer chain to locate the user's current position. Figure 14 (b) nevertheless exhibits lower total cost for our strategy -- with 20~45% reduction -- due to significantly reduced update expense.

Cost comparison between our strategy, the per-user forwarding strategy [11] and the two-level pointer forwarding strategy [10] is given in Figures 15 and 16 (where L represents the pointer chain length of the per-user forwarding strategy; $K1$ and $K2$ are the pointer chain lengths for the two-level pointer forwarding; K is the pointer chain length of our strategy). The result collected under $CMR \leq 2$ and $\delta = 0.3$ in Figure 15 indicates the same tracking cost for our enhanced pointers and the per-user forwarding (because both strategies use the same tracking procedure to locate users). As to the total cost, our enhanced pointers are shown to be more cost efficient (though with only minimal degrees) than the per-user forwarding. Figure 16 presents cost comparison between our strategy and the two-level pointer forward-

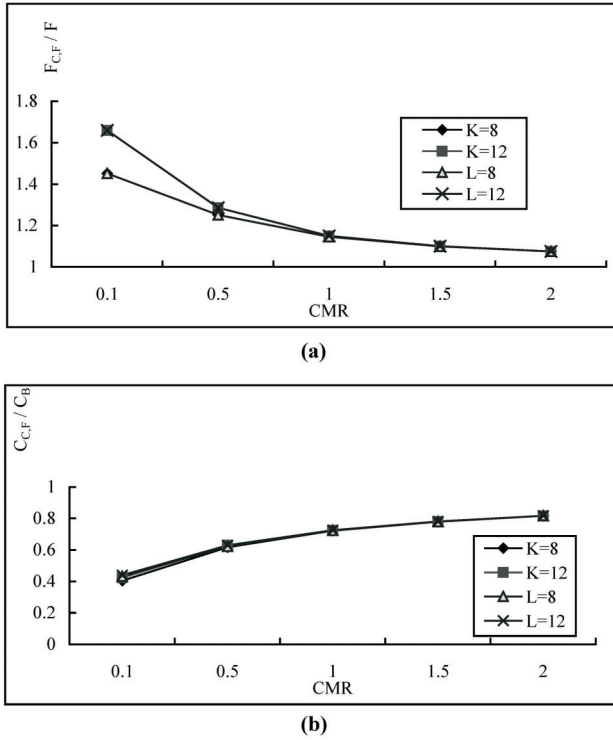


Figure 15. The FIND and total costs at *cache miss* for our strategy and the per-user forwarding scheme with $\delta = 0.3$ and $CMR \leq 2$. (a) the FIND cost F_{CF}/F . (b) the total cost C_{CF}/C_B .

ing strategy under $CMR \leq 2$ and $\delta = 0.3$. The result displays slightly higher tracking cost and total cost for our enhanced caching strategy. This is because the two-level pointer forwarding strategy involves a more complicated operation (using two pointer parameters) than ours to optimize the performance of location tracking. (Note that here we discuss only the cache miss cases. For other cases, our strategy is shown to yield more favorable performance.)

In *cache hit with invalid cache records* cases

Cost comparison between the IS-41 scheme, the enhanced forwarding pointers in our strategy, the per-user forwarding pointers and the two-level forwarding pointers under *cache hit with invalid cache records* is plotted in Figures 17 and 18. Figure 17 exhibits lower FIND cost and total cost for our strategy than for the IS-41 scheme due to our enhanced forwarding pointers. The enhanced forwarding pointers in our strategy are also shown in Figure 18 to have lower FIND cost and total cost than both the per-user forwarding and two-level pointer forwarding strategies. This is because users in our strategy save more HLR query and update cost when frequently crossing RAs' boundaries (recall that our strategy involves

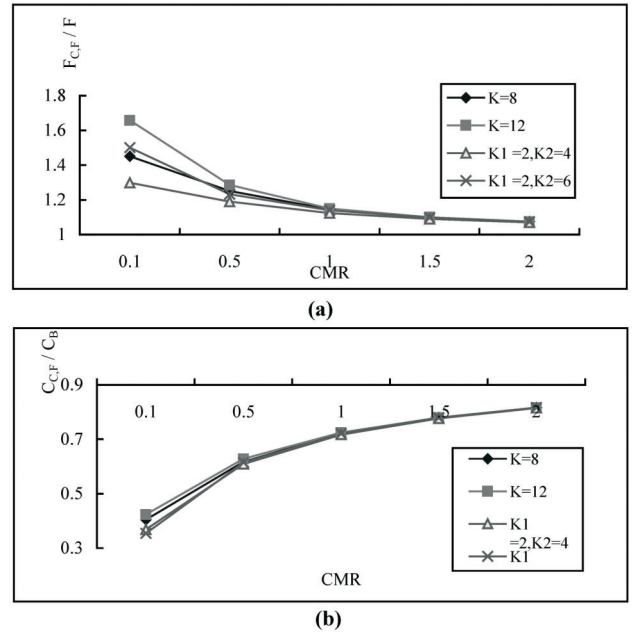


Figure 16. The FIND and total costs at *cache miss* for our strategy and the two-level pointer forwarding scheme with $\delta = 0.3$ and $CMR \leq 2$. (a) the FIND cost F_{CF}/F . (b) the total cost C_{CF}/C_B .

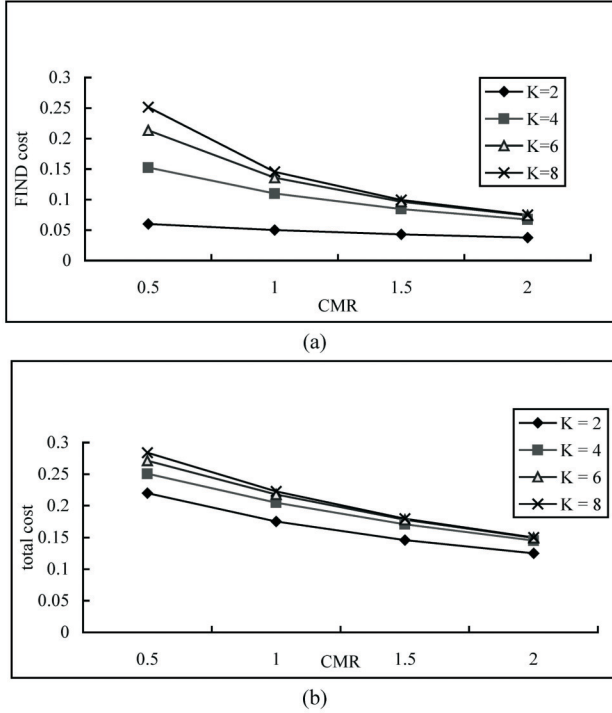


Figure 17. The FIND and total costs at *cache hit with invalid cache records* for our strategy and the IS-41 scheme with $\delta = 0.3$, $\text{CMR} \leq 2$. (a) the traversal cost $F_{C,F}/F$ (b) the total cost $C_{C,F}/C_B$.

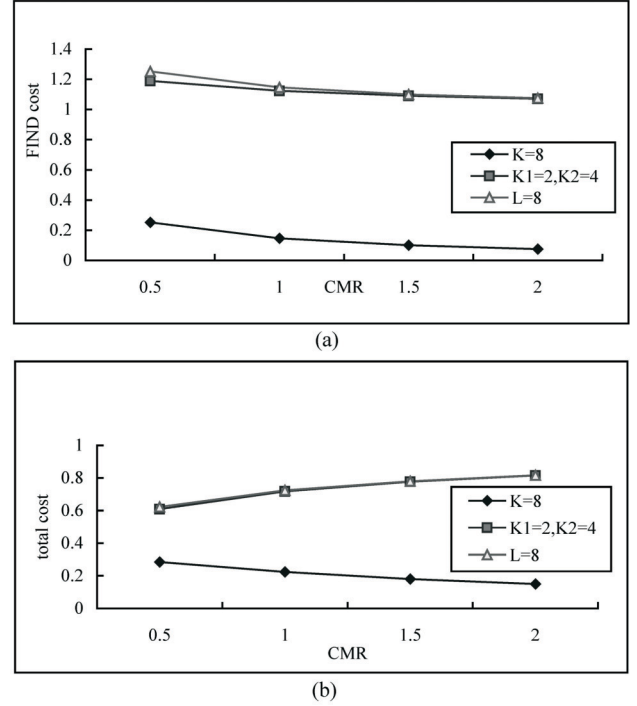


Figure 18. The FIND and total costs at *cache hit with invalid cache records* for our strategy, the per-user pointer scheme and the two-level pointer forwarding scheme with $\delta = 0.3$ and $\text{CMR} \leq 2$. (a) the FIND cost $F_{C,F}/F$ (b) the total cost $C_{C,F}/C_B$.

only the cost of setting up and traversing forwarding pointers, not the cost of registering to and querying HLRs).

In cache hit with valid cache records cases

Note that there is no cost for our enhanced forwarding pointers at *cache hit with valid cache records* because in this situation the user can be located without involving the enhanced pointers. The case is thus removed from our cost comparison analysis. If it is counted in, the result will turn out even more favorable for our strategy.

5. Conclusion

This paper presents an enhanced caching strategy for (user) location tracking in the PCS network. The new strategy enhances the existing caching strategy with forwarding pointers to locate users who make frequent moves while using PCS. Our strategy is able to reduce the cost of registration as it uses forwarding pointers to locate a user when he moves to a new registration area, instead of making queries to the HLR to update the

user's location information. On call delivery, we make use of two features -- users tend to revisit previous RAs and to visit previously called RAs -- to update the cached records and thus lift up the cache hit and valid ratios. The forwarding pointers (NEXT_VLRs in the cache entries) in our strategy also cut down the penalty due to cache hit with invalid cache records, further reducing the communication cost.

Analytic evaluation is conducted to compare different probabilities (*cache hit with valid cache records*, *cache hit with invalid cache records* and *cache miss*) for various caching structures. Invalid cache records tend to cause location miss and extra cost penalty (such as HLR queries) to caching strategies, but not to our enhanced caching strategy. Our strategy, which uses enhanced pointers (NEXT_VLRs) in cache entries to locate users, can achieve location hit and thus reduce the penalty of location miss. Analytical results show that our enhanced caching strategy generates higher location hit probabilities than the other caching strategy under any CMRs. (At lower CMRs, the performance turns out even better – the

location hit probabilities get lifted up to 70%.) It is also shown to increase the probabilities of *cache hit with valid cache records* by using the user's mobility feature of visiting previously called RAs to update cache records.

Employing forwarding pointers in cache entries to locate users is desirable as it helps avoid unnecessary HLR queries and updates occurring in existing pointer forwarding strategies. Compared with other location tracking strategies, such as the IS-41 scheme, the per-user forwarding strategy and the two-level pointer forwarding strategy, our enhanced caching strategy proves to be more efficient in terms of registration cost, traversal cost and total cost. There is indeed no cost for our enhanced forwarding pointers at *cache hit with valid cache records* because in this situation our strategy can locate the user without involving the enhanced pointers.

Acknowledgement

A preliminary version of this paper was presented at the 2004 Int'l Conf. on Embedded and Ubiquitous Computing, Aug. 2004, and the 19th Int'l Conf. on Advanced Information Networking and Applications, Mar. 2005.

This work was supported in part by the National Science Council, Taiwan, R. O. C., under Grant No. NSC 92-2213-E-032-011.

References

- [1] Jain, R., Lin, Y.-B., Lo, C. and Mohan, S., "A Caching Strategy to Reduce Network Impacts of PCS," *IEEE J. Select. Areas Commun.*, Vol. 12, pp. 1434–1444 (1994).
- [2] Lin, Y.-B., "Determining the User Locations for Personal Communications Networks," *IEEE Trans. Veh. Technol.*, Vol. 43, pp. 466–473 (1994).
- [3] Harjono, H., Jain, R. and Mohan, S., "Analysis and Simulation of a Cache-Based Auxiliary User Location Strategy for PCS," *IEEE Conf. Networks Pers. Commun.* (1994).
- [4] EIA/TIA, "Cellular Radio Telecommunications Inter-System Operations," *Technical Report IS-41 (Revision B)*, EIA/TIA (1991).
- [5] Wilson, D. R., "Signaling System No. 7, IS-41 and Cellular Telephony Networking," *Proc. IEEE*, Vol. 80, pp. 652–664 (1992).
- [6] Cox, D. C., "Wireless Personal Communications: What is it?," *IEEE Personal Communication Magazine*, pp. 20–35 (1995).
- [7] Lycksell, E., "GSM System Overview," *Technical Report*, Swedish Telecom. Admin., Jan. (1991).
- [8] Mouly, M. and Pautet, M. B., *The GSM System for Mobile Communications*, France: Palaiseau (1992).
- [9] Mohan, S. and Jain, R., "Two User Location Strategies for Personal Communications Services," *IEEE Pers. Commun.*, pp. 42–50, First Quarter (1994).
- [10] Ma, W. and Fang, Y., "Two-Level Pointer Forwarding Strategy for Location Management in PCS Networks," *IEEE Trans. Mobile Computing*, Vol. 1, pp. 32–45 (2002).
- [11] Jain, R. and Lin, Y.-B., Charles Lo and Seshadri Mohan, "A Forwarding Strategy to Reduce Network Impacts of PCS," *IEEE J. Select. Areas Commun.*, Vol. 15 (1995).
- [12] Sue, K.-L. and Tseng, C.-C., "One-Step Pointer Forwarding Strategy for Location Tracking in Distributed HLR Environments," *IEEE J. Select. Areas Commun.*, Vol. 15, pp. 1455–1466 (1997).
- [13] Pollini, G. P., I, C.-L., "A Profile-Based Location Strategy and Its Performance," *IEEE J. Select. Areas Commun.*, Vol. 15 (1997).
- [14] Mao, Z., Douligeris, C., "An Integrated Strategy for Location Tracking in Mobile Networks," *IEEE Conf. Networks Commun.*, pp. 777–781 (2000).
- [15] Jain, R. and Lin, Y.-B., "An Auxiliary User Location Strategy Employing Forwarding Pointers to Reduce Network Impacts of PCS," *ACM-JCM Wireless Networks*, pp. 197–210 (1995).
- [16] Lin, Y.-B., "Reducing Location Update Cost in a PCS Network," *IEEE/ACM Transaction on Networking*, Vol. 5 (1997).
- [17] Guan, H., Du, Z. T. and Li, C. S., "Terminal Aided K-Location Tracking with Distribute HLR Database for Mobility Management," *IEEE VTC 2000*, pp. 2310–2314.
- [18] Lin, Y.-B., "Location Tracking with Distributed HLR's

- and Pointer Forwarding,” *Proc. 1995 Int’l Symp. Commun.*, pp. 31–37 (1995).
- [19] Weng, C.-M. and Chu, C.-H., “K-Step Pointer Forwarding Strategy for Location Tracking in Distributed HLR Environment,” *IEE Proc. Commun.*, Vol. 150 (2003).
- [20] Chen, I.-R. and Gu, B. “Quantitative Analysis of a Hybrid Replication with Forwarding Strategy for Efficient and Uniform Location Management in Mobile Wireless Networks.” *IEEE Trans. Mobile Computing*, Vol. 2, pp. 3–15 (2003).

Manuscript Received: Oct. 13, 2006

Accepted: Jan. 25, 2008