Two-Level Pointer Forwarding Strategy for Location Management in PCS Networks

Wenchao Ma, *Student Member*, *IEEE*, and Yuguang Fang, *Senior Member*, *IEEE*

Abstract—For a PCS network to effectively deliver services to its mobile users, it must have an efficient way to keep track of the mobile users. The location management fulfills this task through location registration and paging. To reduce the signaling traffic, many schemes such as Local Anchor (LA) scheme, per-user caching scheme and pointer forwarding scheme have been proposed in the past. In this paper, we present a new location management scheme which intends to mitigate the signaling traffic as well as reduce the tracking delay in the PCS systems. In this strategy, we choose a set of VLRs traversed by users as the Mobility Agents (MA), which form another level of management in order to make some registration signaling traffic localized. The idea is as follows: Instead of always updating to the HLR, which would become the bottleneck otherwise, many location updates are carried out in the mobility agents. Thus, the two-level pointer forwarding scheme is designed to reduce the signaling traffic: Pointers can be set up between VLRs as the traditional pointer forwarding scheme and can also be set up between MAs. The numerical results show that this strategy can significantly reduce the network signaling traffic for users with low CMR without increasing much of the call setup delay.

Index Terms—PCS networks, mobility management, pointer forwarding, location management.

1 Introduction

Personal Communications Systems (PCS) can provide wireless communication services to users on the move. To deliver services more effectively to a mobile user, it is important to have an efficient way to locate the mobile user [7]. Location management is used to track mobile subscribers in the PCS networks. Two standards currently exist for PCS location management: IS-41 [1] and GSM MAP [2]. The IS-41 is commonly used in North America, while the GSM is popular in Europe and Asia. Both strategies share the same characteristics, they both use a two-tier system of Home Location Register (HLR) and Visitor Location Register (VLR) databases. According to these strategies, a mobile user performs location update (registration) at the HLR every time the user crosses the boundary of a Registration Area (RA) and deregisters at the previous VLR. Thus, the registrations will incur high signaling traffic if many users update their locations to the HLR. If many users are far away from their HLRs, heavy signaling traffic over the network can occur. This problem becomes more serious with the increase of the number of mobile users. Many research works have been carried out to overcome the problem [6]. The local anchor (LA) scheme, proposed by Ho and Akyildiz [3], reduces the signaling traffic by using a local anchor (a VLR a mobile user is currently visiting when he/she receives a call). In this scheme, a VLR close to the user is selected as the local anchor (LA) for the user. Whenever a user moves from one RA to another, it will perform location update to the LA. A LA for a mobile will change only when a call request to the mobile arrives; at the same time, the HLR is also updated via the registration process. When a call request terminating at this user is received by the HLR, the user can be traced to the LA. The LA scheme avoids update to HLR completely at the expense of the increase in local signaling traffic. The drawback of this scheme is that when the user keeps moving constantly without receiving any call, the updates to LA may become costly, a similar bottleneck as the HLR. For example, at the end of conferences or games, many people move away from one site without receiving calls, the LA for these people can become bottleneck. Jain and Lin proposed another scheme called per-user pointer forwarding scheme [4]. In this scheme, some updates to the HLR can be avoided by setting up a forwarding pointer from the previous VLR to the new VLR. When a call request to a mobile user arrives, the PCS network first queries the user's HLR to determine the VLR, which the user was visiting at the previous location update, then follows a chain of forwarding pointers to the user's current VLR to find the mobile user. The traffic to the HLR is decreased by using the pointer chain; however, the penalty is the time delay for tracking the user when a call to the user arrives. The longer the pointer chain, the less the signaling traffic, the longer the setup delay for finding the user. To avoid long setup delay, a threshold of the length of the pointer chain is used. The user needs to perform registration to the HLR after the chain threshold is reached. In order to overcome the drawbacks of the above two schemes, we propose a two-level pointer forwarding strategy. Two kinds of pointers are used in this scheme. Some VLRs are selected as the Mobility Agents (MA), which will be responsible for location management in a larger area comparing to the RAs and can be geographically distributed. They are not only managing the location information of mobile users traveling in their coverage areas, but also providing other services (such as serving as Foreign Agents in mobile IP networks, intrusion detection devices). Thus, all MAs can form what we call the *Virtual Management Network (VMN)*. In this paper, we concentrate only on the

- *The authors are with the Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building, PO Box 116130, Gainesville, Florida 32611-6130.*
  E-mail: mawench@ufl.edu and fang@ece.ufl.edu.

Manuscript received 17 Feb. 2001; revised 22 Nov. 2001; accepted 25 Feb. 2002.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 10-022002.

1536-1233/02/$17.00 © 2002 IEEE

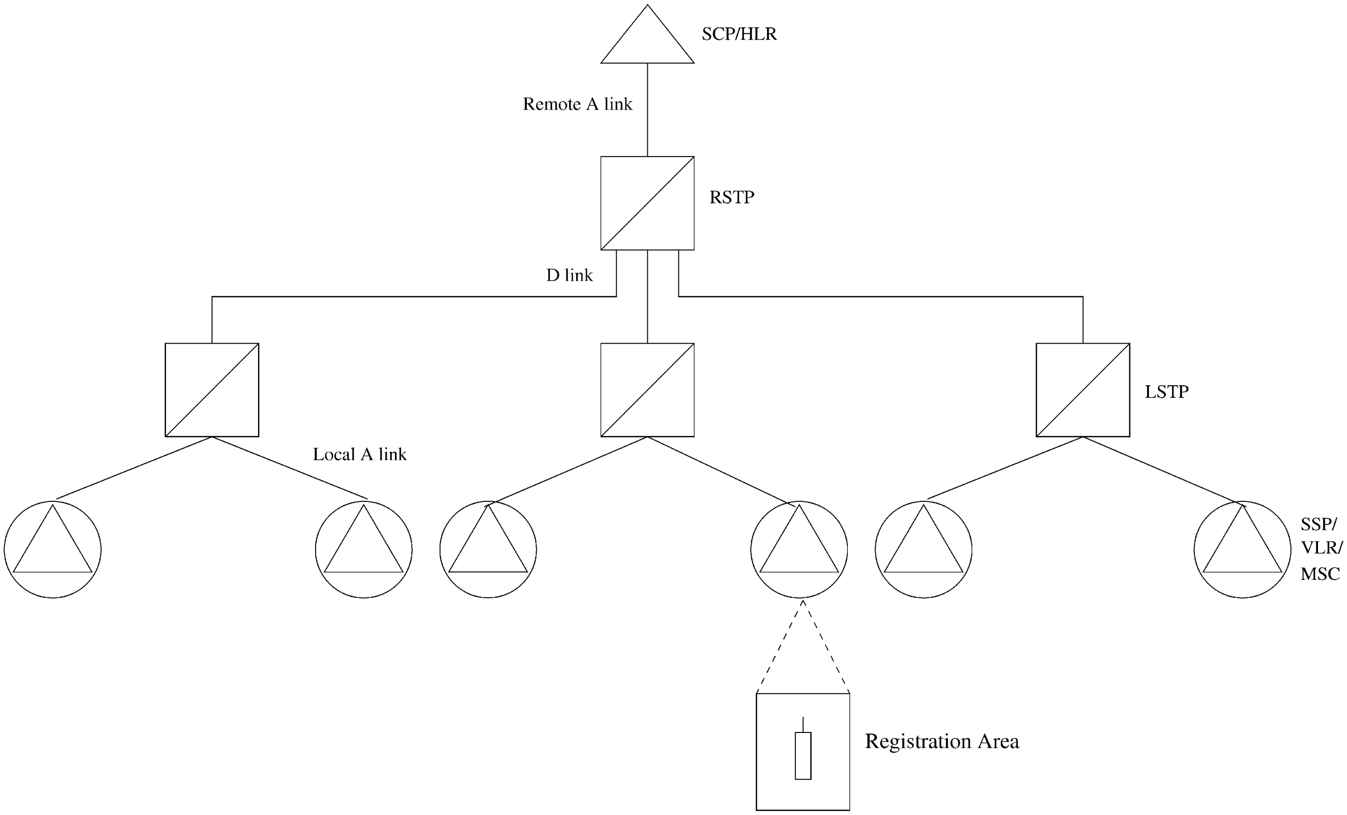


Fig. 1. Reference CCS network architecture.

location management purpose. The pointers between MAs are level_1 pointers and those between VLRs in the same charging domain of MAs are level_2 pointers. When the user crosses the boundaries of RAs, the level_2 pointers are set. If the level_2 pointer chain threshold is reached, the current RA is selected as an MA for this user and a level_1 pointer is set up from previous MA. Calls to a given user will query the HLR first and follow the level_1 pointer chain to the current MA, then reach the user current VLR by tracking the level_2 pointer chain. The user does not need to update the HLR until the level_1 pointer chain threshold is reached. The chain threshold in the two-level pointer forwarding strategy can be much longer than that in the simple pointer forwarding scheme, but can have shorter call setup delay due to the level_1 pointer chain. The two-level pointer forwarding scheme can avoid the possible costly updates to HLR and the bottleneck of local anchor. More importantly, the thresholds for the pointer chains are two parameters which can provide the flexibility in design comparing to the one-parameter traditional pointer forwarding strategy. Finally, we point out that our proposed scheme can be easily modified for the future generation wireless systems [9], which is briefly discussed in this paper. More comprehensive research along this line will be carried out in the future.

This paper is organized as follows: In Section 2, we describe the basic PCS network architecture to facilitate the presentation and analysis of the basic and two-level pointer forwarding strategies. Section 3 introduces the basic IS-41 location management and the new two-level pointer forwarding strategy in detail. We analyze the performance of the new scheme and compare it with the basic IS-41 scheme, theoretically, in Section 4, we also analyze how the user RA residence time can affect the two-level forwarding scheme performance in this section. We then compare the performance of the two-level pointer forwarding strategy with local anchor and per-user forwarding schemes in Section 5. In Section 6, we briefly discuss the relationship between our new two-level pointer forwarding scheme and the previous schemes, such as the per-user forwarding scheme, and present some alternative ways to select Mobility Agents and the implementation issues of the new scheme in 3G wireless communication systems. Section 7 provides the conclusions.

2 PCS Network Architecture

In a PCS networks, the service area covered by the PCS network is divided into cells. Each cell is primarily served by one base station, although a base station may serve one or more cells. An RA consists of an aggregation of a number of cells, forming a contiguous geographical region. The signaling network used to set up calls is distinct from the network used to actually transport the information contents of the calls. Specially, we assume that a Common Channel Signaling (CCS) network is used to set up calls which use the Signaling System No.7 (SS7) protocols (see Fig. 1). All the base stations in an RA are connected via a wire-line network to an end-office switch or Service Switching Point (SSP). Each SSP serves an RA. All the SSPs of different RAs are, in turn, connected to a higher hierarchical Local Signaling Transfer Points (LSTP), which are connected to a Regional STP (RSTP). An RSTP connects to all the LSTPs in one region. In practice, however, each STP actually consists of two STPs in a mated-pair configuration for reliability; for the simplicity of presentation, Fig. 1 only shows one of each pair. The RSTPs are also connected to a Service Control Point (SCP). Each SCP is equipped with a HLR database. For simplicity, we assume that each VLR is associated with one Mobile Switching Center (MSC), which connects the BSs and backbone communication infrastructure (such as Public Switched Telephone Network (PSTN)). Therefore, we assume that an MSC, an SSP, and a VLR database are associated together to serve an RA. The configuration may vary in practice, however, the assumption is used only for performance analysis. Since we do not deal with the content of the messages, we assume that the message sizes are equal for all signaling transactions. We will compare the cost of the basic strategy with and without the two-level forwarding scheme in terms of signaling traffic.

3 Two-Level Pointer Forwarding Strategy

To facilitate the presentation, the following two operations are defined

1. *MOVE*: the PCS user moves from one RA to another and
2. *FIND*: determination of the RA where the PCS user is currently located.

Next, we present the procedures for the location management schemes used in IS-41 and our new strategy.

3.1 Basic User Location Management Scheme in IS-41

We call the *MOVE* and *FIND* used in current PCS standards, such as IS-41 or GSM MAP, the *BasicMOVE* and *BasicFIND*. We present the procedures in the following pseudocode. We remark that the *BasicMOVE* and *BasicFIND* procedures we present here are simplifications of those in the standards; however, such simplifications do capture the major interactions between the HLR and VLR databases relevant to our comparative study.

```
BasicMOVE()
  {
     The mobile terminal detects that it is in a new
        registration area;
     The mobile terminal sends a registration message to the
        new VLR;
     The new VLR sends a registration message to the user's
        HLR;
     The HLR sends a registration cancellation message to
        the old VLR;
     The old VLR sends a cancellation confirmation message
        to the HLR;
     The HLR sends a registration confirmation message to
        the new VLR;
  }
BasicFIND()
  {
     Call to a PCS user is detected at the local switch;
     if the called party is in the same RA, then return;
     Switch queries the called party's HLR;
     HLR queries the called party's current VLR,$V$;
     VLR V returns the called party's location to HLR;
     HLR returns the location to the calling party;
  }
```

3.2 Two-Level Pointer Forwarding Scheme

The two-level pointer forwarding procedures modify the basic procedures as follows: When a user moves from one RA to another, it informs the switch (and the VLR) at the new RA about the old RA. It also informs the new RA about the previous MA it was registered to. The switch at the new RA determines whether to invoke the *BasicMOVE* or the *TwoLevelFwdMOVE* strategy.

In *TwoLevelFwdMOVE()*, the new VLR exchanges messages with the old VLR or the old MA to set up a forwarding pointer from the old VLR to the new VLR. If a pointer is set up from the previous MA, the new VLR is selected as the current MA. The *TwoLevelFwdMOVE* procedures do not involve the user's HLR. Fig. 2 shows *Two-Level Forward MOVE* procedures with level_1 pointers chain threshold limited to three. Assume that a user moves from RAa to RAg (these RAs are not necessary to be adjacent) and RAa is the user's MA. When the user leaves RAa but before enters RAb, the user informs the new VLRs and the level_2 pointers are built from the old VLR to the new VLR. When the user enters RAb, the chain threshold for level_2 pointer is reached, so RAb is selected as the user's new MA and a level_1 pointer is set up from the old MA to the new MA. At the same time, level_2 pointer chain is reset. The similar procedures are used at RAc. A level_1 pointer is set up from RAb to RAc and the VLR in RAc is the user's new MA. As the user keeps moving. In RAe, the threshold of a level_2 pointer chain is reached again, while this time the threshold of the level_1 pointer chain is reached, too. Instead of exchanging information with the previous MA, the *BasicMOVE()* is invoked. The HLR is updated with the user current location. The messages REGPTR_L1 and REGPTR_L2 are messages from the new VLR to the old VLR specifying that a level_1 or level_2 forwarding pointer is to be set up; messages regptr_l1 and regptr_l2 are the confirmations from the old VLR (or MA). In this figure, the VLRs in RAa, RAb, RAe, and RAf are selected as the user's MAs.

The *TwolevelFwdFIND()* procedure is invoked for the subsequent calls to the user from some other switches. The user's HLR is queried first as in the basic strategy, and a pointer to the user's potentially outdated MA is obtained. The pointer chain is followed to find out the user's current location (see Fig. 3). As we can see, in the two-level pointer forwarding scheme, the chain length can be longer than that in the basic pointer forwarding scheme without increasing the *Find* penalty significantly. The previous study [4] shows that more saving can be obtained with longer chain. However, the pointer chain length is limited by the delay restriction requirement. By appropriately tuning the two thresholds in our schemes, we can mitigate the signaling cost without too much increase in call setup delay.

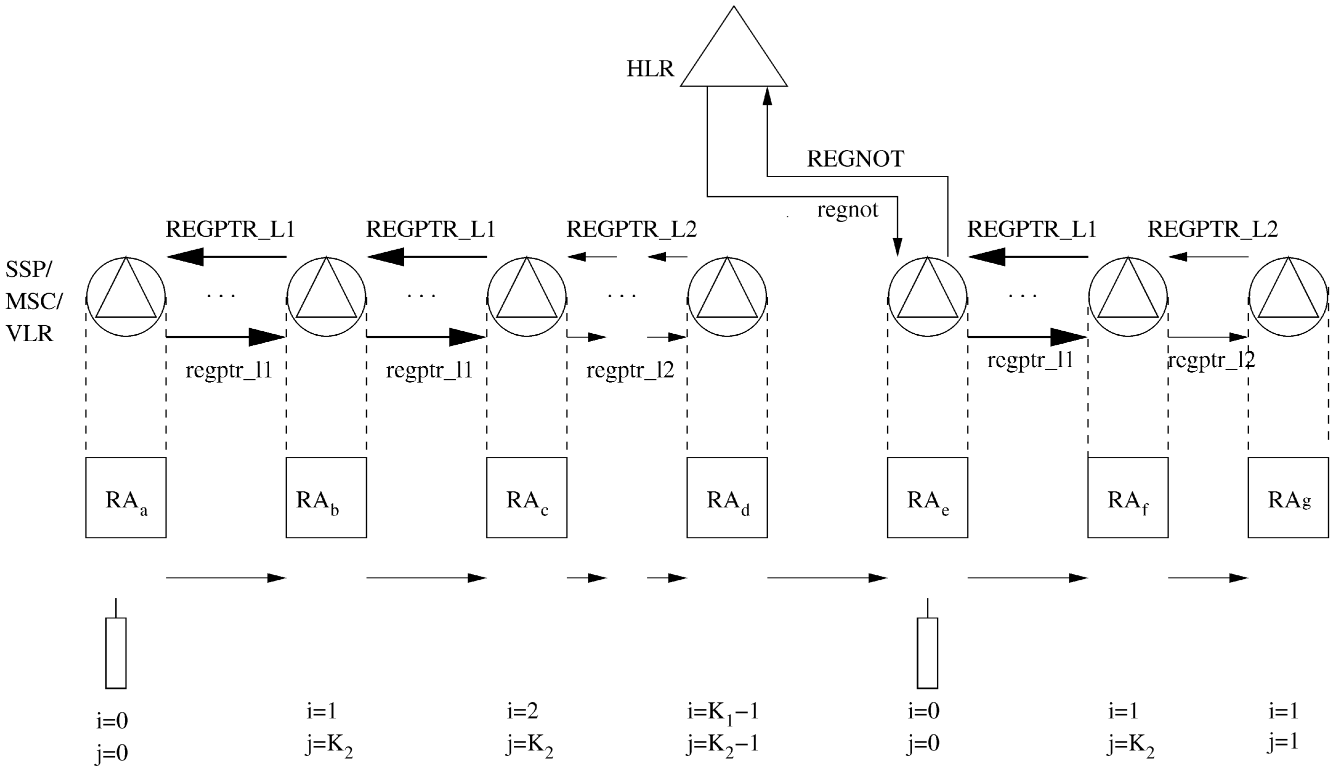


Fig. 2. The *TwoLevelFwdMOVE()* procedure.

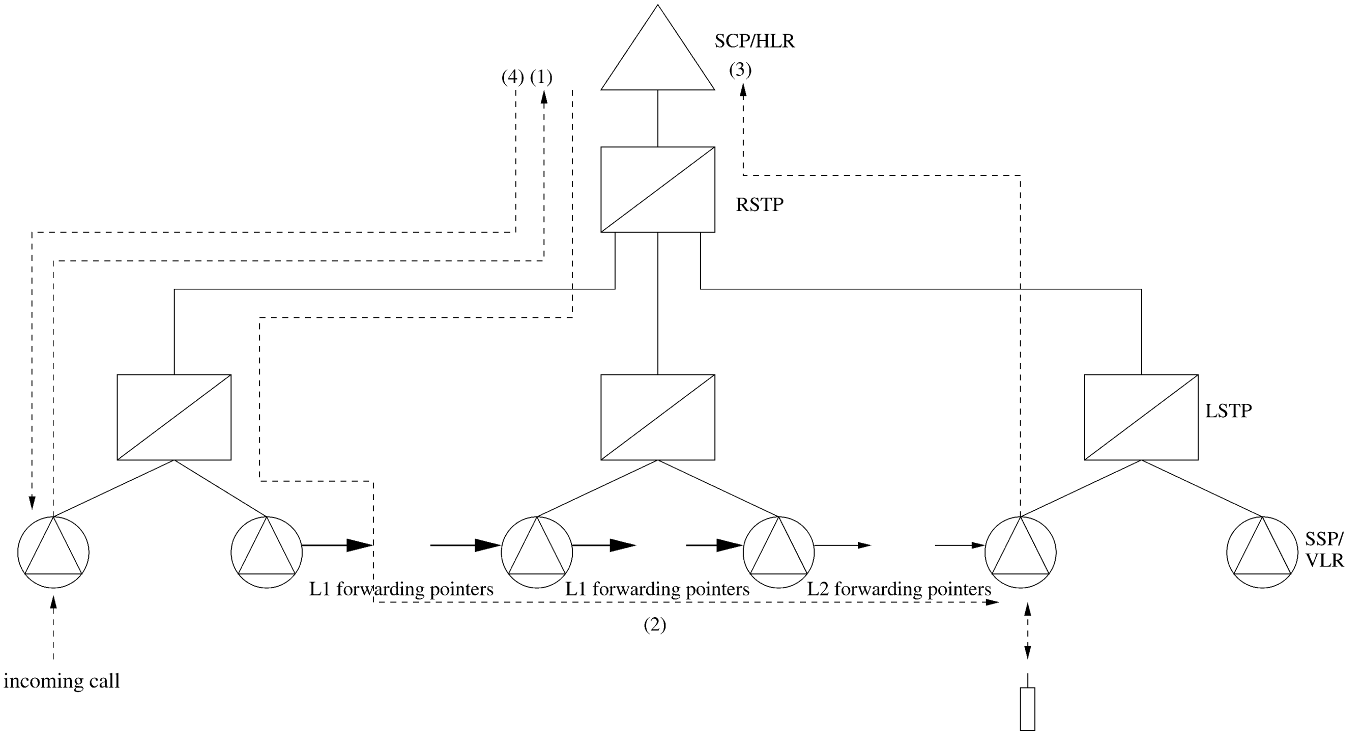


Fig. 3. The *TwoLevelFwdFIND()* procedure.

We describe the two-level pointer forwarding procedures using the following pseudocode (We use the shared global variable i and j in the pseudocode).

```
TwoLevelFwdMOVE()
  {/*Initially, $i, j$ are 0*/
  if($j < K_2$ and $i < K_1$)    {
The user registers at the new RA/VLR, passing the id of
      the former RA/VLR and MA;
The new VLR deregisters the user at old VLR;
The old VLR sends ACK and the user's service profile to
      the new VLR;
$j := j + 1$;    }
```

```
  else if($j >= K_2$ and $i < K_1$)  {
    The user registers at the new RA/VLR(MA), passing the id of the former RA/VLR and MA;
    The new VLR deregisters the user at the old MA/VLR;
    The old MA/VLR sends ACK and the user's service profile to the new MA/VLR;
    $i := i + 1;\ j := 0;$    }
  else  {
 BasicMOVE();
 $i := 0;\ j := 0;$    }
  }
TwoLevelFwdFIND()
  {
    A call to the PCS user is detected at a local switch;
    if the called party is in the same RA, then return;
    The local switch queries the called party's HLR;
    HLR queries $V_0$/MA;
    While(Queried VLR is not the called party's current VLR);
    VLR queries the next VLR in the pointer chain;
    /*Now the called party's actual VLR has been found*/
    $i := 0;\ j := 0;$
    The called party's current VLR sends the user location to HLR;
    HLR sends the user location to calling party's switch;
  }
```

4 Performance Analysis

In this section, we develop an analytic model and study the performance of two-level pointer forwarding strategy based on different parameters for different classes of users.

We characterize the classes of users according to their call-to-mobility ratio (CMR). The CMR of a user is defined as the expected number of calls to a user during the period that the user visits an RA (Notice that the CMR is defined here in terms of the calls received by a particular user, not calls originated from the user). If calls are *received* by the user at a mean rate λ and the time the user resides in a given RA has a mean $1/\mu$, then the CMR, denoted as p, is given by

$$p = \lambda/\mu. \tag{1}$$

In order to make comparison of the cost with that of the basic IS-41 location management scheme, we need to model the basic procedures used in IS-41. Assume that a user crosses several RAs between two consecutive calls. If the basic user location strategy is used, the user's HLR is updated every time the user moves to a new RA. If the two-level pointer forwarding strategy is used, the HLR is updated only every $K_1 \cdot K_2$ moves (K_1 and K_2 are the level_1 and level_2 pointer chain length threshold, respectively), while forwarding pointers are set up for all other moves.

We define C_B and C_F to be the total costs of maintaining the location information (location updating) and locating the user (location tracking) between two consecutive calls for the basic strategy and the two-level forwarding strategy, respectively. The following notations will be used in our analysis:

- $m =$ the cost of a single invocation of *BasicMOVE*.
- $M =$ the total cost of all the *BasicMOVESs* between two consecutive calls.
- $F =$ the cost of a single *BasicFIND*.
- $M' =$ the expected cost of all *TwoLevelFwdMOVEs* between two consecutive calls.
- $F' =$ the average cost of the *TwoLevelFwdFIND*.
- $S_1 =$ the cost of setting up a forwarding pointer (level_1 pointer) between MAs during a *TwoLevelFwdMOVE*.
- $S_2 =$ the cost of setting up a forwarding pointer (level_2 pointer) between VLRs during a *TwoLevelFwdMOVE*.
- $T_1 =$ the cost of traversing a forwarding pointer (level_1 pointer) between MAs during a *TwoLevelFwdFIND*.
- $T_2 =$ the cost of traversing a forwarding pointer (level_2 pointer) between VLRs during a *TwoLevelFwdFIND*.
- $K_1 =$ the threshold of level_1 pointer chain.
- $K_2 =$ the threshold of level_2 pointer chain.
- $\alpha(i) =$ the probability that there are i RA crossings between two consecutive calls.

Then, we have

$$C_B = M + F = m/p + F. \tag{2}$$

$$C_F = M' + F'. \tag{3}$$

Now, we can derive formulas for M' and F' as follows: Suppose that a user crosses i RA boundaries between two consecutive calls. The HLR is updated $\lfloor\frac{i}{K_1K_2}\rfloor$ times. There are also $\lfloor\frac{i}{K_2}\rfloor - \lfloor\frac{i}{K_1K_2}\rfloor$ level_1 pointer creations (every K_2 moves may require a level_1 pointer creation but sometimes the HLR is updated and level_1 pointer is not set up). The level_2 pointers are created for all the rest $i - \lfloor\frac{i}{K_2}\rfloor$ moves. Thus, we obtain

$$M' = \sum_{i=0}^{\infty}\left\{\left\lfloor\frac{i}{K_1K_2}\right\rfloor m + \left(\left\lfloor\frac{i}{K_2}\right\rfloor - \left\lfloor\frac{i}{K_1K_2}\right\rfloor\right)S_1 + \left(i - \left\lfloor\frac{i}{K_2}\right\rfloor\right)S_2\right\}\alpha(i). \tag{4}$$

The cost F' is derived as follows: After the last *BasicMove* operations (if any), the user traverses

$$\left\lfloor\frac{i - \left\lfloor\frac{i}{K_1K_2}\right\rfloor K_1K_2}{K_2}\right\rfloor$$

level_1 pointers and

$$i - \left\lfloor\frac{i}{K_1K_2}\right\rfloor K_1K_2 - \left\lfloor\frac{i - \left\lfloor\frac{i}{K_1K_2}\right\rfloor K_1K_2}{K_2}\right\rfloor K_2$$

level_2 pointers. Thus, we obtain

$$
\begin{aligned}
F' = F + \sum_{i=0}^{\infty} \Bigg\{ & \left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor T_1 \\
& + \left(i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2 \right. \\
& \left. - \left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor K_2 \right) T_2 \Bigg\} \alpha(i).
\end{aligned} \tag{5}
$$

In order to evaluate $\alpha(i)$, we make the following assumptions:

1. The call arrivals to a user form a Poisson process with arrival rate λ.
2. The residence time of a user at a registration area is a random variable with a general density function $f_m(t)$ and the Laplace transform

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

The expected residence time of a user at an RA is $1/\mu$.

We denote $g = f_m^*(\lambda)$ for convenience. With these assumptions, it can be shown that (see Appendix for the derivation) that:

$$M' = \frac{S_2}{p} + \frac{(1-g)g^{K_2-1}(S_1 - S_2)}{p(1-g^{K_2})} + \frac{(1-g)g^{K_1K_2-1}(m - S_1)}{p(1-g^{K_1K_2})}, \tag{6}$$

$$
\begin{aligned}
F' = F &+ \frac{[1 - K_1K_2g^{K_1K_2-1} + (K_1K_2 - 1)g^{K_1K_2}]T_2}{p(1-g^{K_1K_2})} \\
&+ \frac{(T_1 - K_2T_2)(1-g)[g^{K_2} - K_1g^{K_1K_2} + (K_1 - 1)g^{(K_1+1)K_2}]}{pg(1-g^{K_1K_2})(1-g^{K_2})}.
\end{aligned} \tag{7}
$$

For demonstration purposes, we assume that the RA residence time of a user is Gamma distributed with mean $1/\mu$. The reason that Gamma distribution is selected is its flexibility 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

$$f_m^*(s) = \left(\frac{\gamma\mu}{s + \gamma\mu} \right)^{\gamma},$$

thus, we have,

$$g = f_m^*(\lambda) = \left(\frac{\gamma\mu}{\lambda + \gamma\mu} \right)^{\gamma} = \left(\frac{\gamma}{p + \gamma} \right)^{\gamma}. \tag{8}$$

In particular, when $\gamma = 1$, we have an exponential distribution for the RA residence time.

4.1 Performance Analysis for Exponential RA Residence Time

We first consider the situation when the RA residence time is exponentially distributed. By setting $\gamma = 1$ in (8), we have

$$g = \frac{1}{1+p}$$

and (6) and (7) can be rewritten as

$$M' = \frac{S_2}{p} + \frac{S_1 - S_2}{(1+p)^{K_2} - 1} + \frac{m - S_1}{(1+p)^{K_1K_2} - 1}, \tag{9}$$

$$
\begin{aligned}
F' = F &+ \frac{T_2}{p} - \frac{T_2K_1K_2}{(1+p)^{K_1K_2} - 1} \\
&+ \frac{(T_1 - K_2T_2)[(1+p)^{K_1K_2} - K_1(1+p)^{K_2} + K_1 - 1]}{[(1+p)^{K_1K_2} - 1][(1+p)^{K_2} - 1]}.
\end{aligned} \tag{10}
$$

From (9), (10), and (3), we obtain

$$
\begin{aligned}
C_F = F &+ \frac{T_2 + S_2}{p} + \frac{S_1 - S_2}{(1+p)^{K_2} - 1} + \frac{m - S_1 - T_2K_1K_2}{(1+p)^{K_1K_2} - 1} \\
&+ \frac{(T_1 - K_2T_2)[(1+p)^{K_1K_2} - K_1(1+p)^{K_2} + K_1 - 1]}{[(1+p)^{K_1K_2} - 1][(1+p)^{K_2} - 1]}.
\end{aligned} \tag{11}
$$

We notice that updating the HLR and performing a *BasicFIND* involve the same number of messages between HLR and VLR databases, so we can choose $m = F$. Without loss of generality, we can normalize $m = 1$. We also assume that the cost of setting up a forwarding pointer is about twice the cost of traversing it since twice as many messages are involved, i.e., we set $S_1 = 2T_1$ and $S_2 = 2T_2$. We consider $S_2 = \delta$ with $\delta < 1$. Since the level_1 pointer is more expensive than level_2 pointer in terms of setup cost, we can assume $S_1 = KS_2$ with $K \geq 1$. It is reasonable to assume that $S_1 < 1$, too. We will see later, however, that the two-level forwarding strategy can also perform well even with $S_1 \geq 1$. From (2), (9), (10), and (11), we obtain,

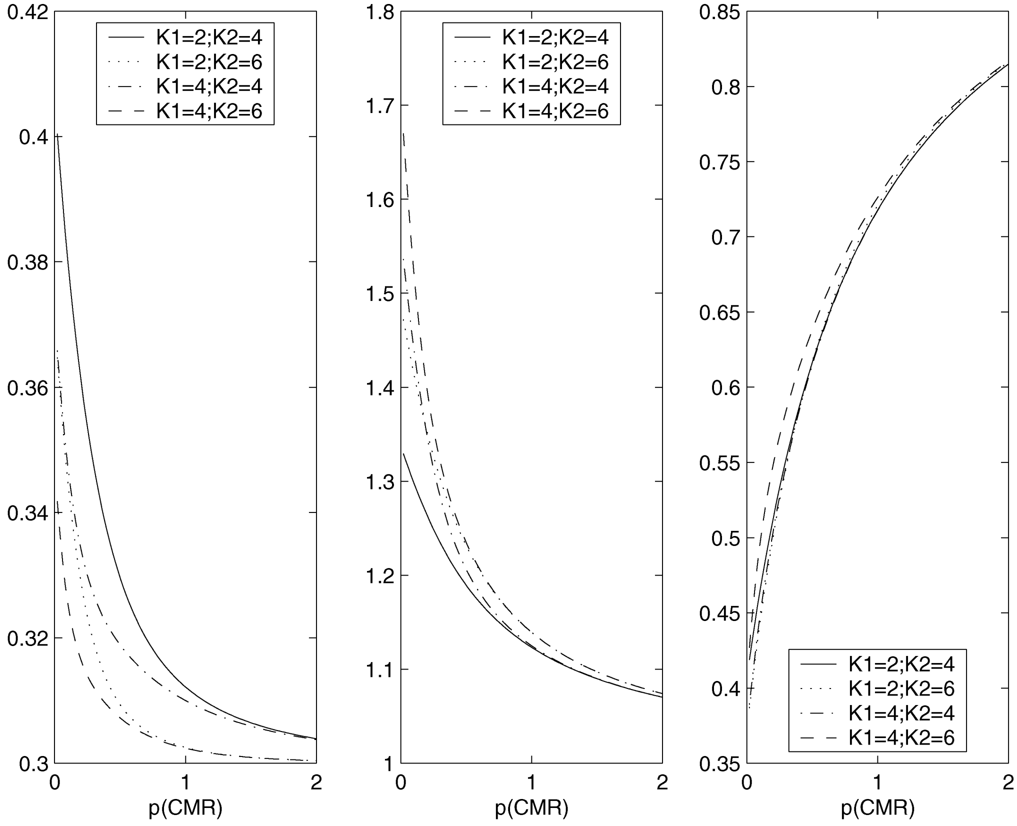
$$C_B = 1 + \frac{1}{p}, \tag{12}$$

$$\frac{M'}{M} = \delta + \frac{(K-1)\delta p}{(1+p)^{K_2} - 1} + \frac{(1 - K\delta)p}{(1+p)^{K_1K_2} - 1}, \tag{13}$$

$$
\begin{aligned}
\frac{F'}{F} = 1 &+ \frac{\delta}{2p} - \frac{K_1K_2\delta}{2[(1+p)^{K_1K_2} - 1]} \\
&+ \frac{\delta(K - K_2)[(1+p)^{K_1K_2} - K_1(1+p)^{K_2} + K_1 - 1]}{2[(1+p)^{K_1K_2} - 1][(1+p)^{K_2} - 1]},
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
\frac{C_F}{C_B} = \frac{p}{1+p} \Bigg\{ 1 &+ \frac{3\delta}{2p} + \frac{(K-1)\delta}{(1+p)^{K_2} - 1} + \frac{1 - (K + \frac{1}{2}K_1K_2)\delta}{(1+p)^{K_1K_2} - 1} \\
&+ \frac{\delta(K - K_2)[(1+p)^{K_1K_2} - K_1(1+p)^{K_2} + K_1 - 1]}{2[(1+p)^{K_1K_2} - 1][(1+p)^{K_2} - 1]} \Bigg\}.
\end{aligned} \tag{15}
$$

In Figs. 4, 5, and 6, we plot the costs as functions of CMR for various values of K_1, K_2, K, and δ. Fig. 4a shows that under certain conditions ($\delta = 0.3, K = 1.5$), two-level forwarding can result in $60 - 70$ percent reductions in location update cost comparing to the basic strategy. However, Fig. 4b indicates that the *FIND cost* of the two-level forwarding scheme is higher than the basic strategy. The reason is that the call for the user needs to traverse the pointer chain to find the user's current location. However,



(a) (b) (c)

Fig. 4. Relative *MOVE* and *FIND* costs of forwarding with $\delta = 0.3$ and $K = 1.5$. (a) The MOVE cost M'/M. (b) The FIND cost F'/F. (c) The net cost C_F/C_B.

as we observe in Fig. 4c, the two-level forwarding strategy can result in a 20-60 percent reduction in the total cost. If we study the plots carefully, we can observe that both the relative *MOVE* an *FIND* cost are decreasing functions of p (CMR). When p is small, the user crosses RAs more frequently. The pointers are needed to be set up and a long chain of pointers have to be traversed, leading to the high setup cost. The improvement of total cost increases when the p decreases because most *MOVEs* do not result in HLR updates but pointer creations. Consider Fig. 4a again, we can have more saving in the *MOVEs* with a longer pointer chain because more updates to HLR can be substituted with pointer creations. However, a long pointer chain increases the *FIND* penalty at the same time (see Fig. 4b). An advantage of the two-level forwarding strategy is that it can have a long pointer chain without increasing the delay penalty significantly because the pointer chain can be shortened by the level_1 pointers between MAs. Under the assumed conditions, the maximum pointer chain length can increase from eight to 24 with only 30 percent *FIND* penalty increase.

Figs. 5a, 5b, and 5c show the plots when K increases from 1.5 to 4. As we can see, even in this case, the cost of setting up a level_1 pointer exceeds the cost of updating HLR, there is only slight increase of the total cost. The *MOVE* and *FIND* costs both increase because the cost of setting up and traversing level_1 pointers chain increases. Since the level_1 pointer is built up only when the level_2 pointer chain threshold is reached and the number of level_2 pointers is dominant, the two-level forwarding strategy is not sensitive to the variation of K. Figs. 6a, 6b, and 6c indicate that the level_2 pointer operation cost δ has more effect on the system performance. In Figs. 6a, 6b, and 6c, δ is increased from 0.3 to 0.6. The *MOVE, FIND* and the net cost all increase. Finally, we can observe that, for small δ, increasing pointer chain length reduces the cost of two-level forwarding scheme (because the pointer operations are cheaper).

4.2 Sensitivity of Performance to Residence Time Variance

We now investigate the sensitivity of the performance costs and benefits of the two-level forwarding scheme to the variance in the user's mobility patterns. We assume that call arrivals to a user form a Poisson process and the RA residence time has a Gamma distribution.

For a Gamma distribution, the variance is $V = \frac{1}{\gamma\mu^2}$. That is, a large γ implies a small variance. Fig. 7 shows the effect of γ on $M'/M, F'/F$ and C_F/C_B. In Fig. 7, we observe that the increase of the variance of RA residence time (smaller γ) causes the increase of M'/M but the decrease of F'/F; the net effect to C_F/C_B is not significant.

Consider M'/M for $\gamma < 1$ comparing to the case when $\gamma = 1$. For a given $p > 0$ (see Fig. 7a), the large variance of RA residence time implies high variation of the RA

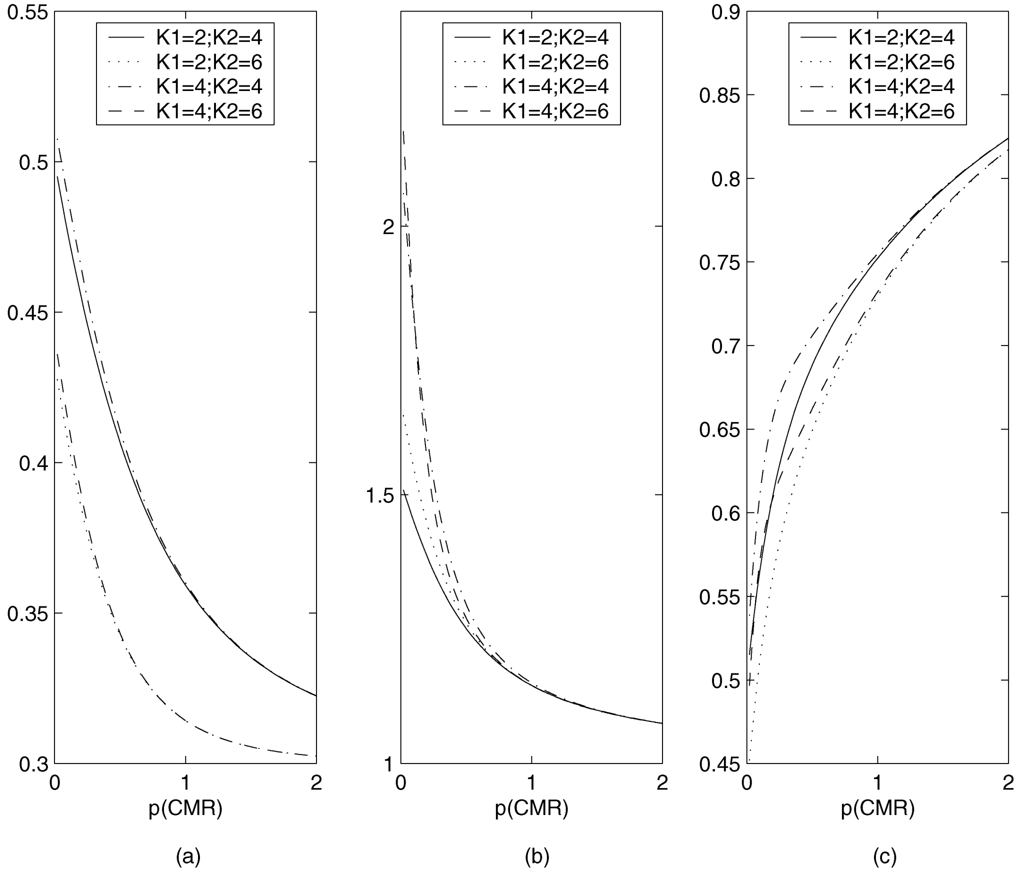


Fig. 5. Relative *MOVE* and *FIND* costs of forwarding with $\delta = 0.3$ and $K = 4$. (a) The MOVE cost M'/M. (b) The FIND cost F'/F. (c) The net cost C_F/C_B.

boundary crossing patterns. If the user crosses many RAs, a longer pointer chain will be created. When the chain limit $K_1 \cdot K_2$ reaches, the HLR will be updated, resulting in increase of M'. On the other hand, if fewer boundaries are crossed, only shorter pointer chains will be set up and the pointer creation/tracing cost will be saved. The net effect is an increase in M'. Now, consider the effect of variance in the RA residence time to the F'/F (See Fig. 7b). When the variance is high, the number of RA boundaries the user crosses between two consecutive calls will vary greatly. When the number is small, the *FIND* cost is reduced; when the number is large, the pointer chain could be shortened. The net effect is a significant improvement in F'/F. The net effect of the variance of the residence time on total cost ratio C_F/C_B is not significant for low CMR (see Fig. 7c).

5 Performance Comparisons

In this section, we compare the performance of the two-level pointer forwarding algorithm with those of per-user forwarding and local anchor schemes.

One of the advantages of the two-level pointer forwarding strategy is that it can have a long pointer chain without much increase of the *FIND* delay. The longer the pointer chain is, the fewer updates to the HLR. Although the net cost cannot be improved greatly with a long pointer chain (Fig. 8), the frequent updates to the HLR can be avoided by a long pointer chain. Let L denote the pointer chain length threshold of a per-user forwarding scheme. In our analysis, we assume that the cost of transmitting and receiving a signaling message between MSC and HLR is a fixed value. In practice, the cost could vary or be higher as the signaling trunks or HLRs become congested. In that case, the two-level pointer forwarding strategy will be more efficient. In Fig. 8b, we can observe that, for the same pointer chain length, the tracking delay in the two-level pointer forwarding scheme to find the user's current location is less than that in the per-user forwarding scheme.

In order to see this more clearly, we plot the relative finding delay for these two schemes in Fig. 9. The relative delay here is defined as the ratio of the finding delay for the two-level pointer forwarding scheme (the proposed scheme) to that for the per-user pointer forwarding scheme. In Fig. 9, we assume that the signaling message will travel back and forth along the same route from HLR to the user's current VLR and that the traversing delay for the level_1 pointer is 1.5 times of that for the level_2 pointer. As we can see from Fig. 9, when the CMR is less than 1, the delay in the two-level pointer forwarding scheme is less than that in per-user forwarding strategy. The effect is more obvious when the pointer length is longer. Since the thresholds for pointer chain are two parameters in our scheme, it is more flexible for the system operator to select different strategies for different users. We observe from the curves, for pointer chain length 12, that the delay is less for $K_1 = 3, K_2 = 4$ than that for $K_1 = 2, K_2 = 6$. The reason is that the system

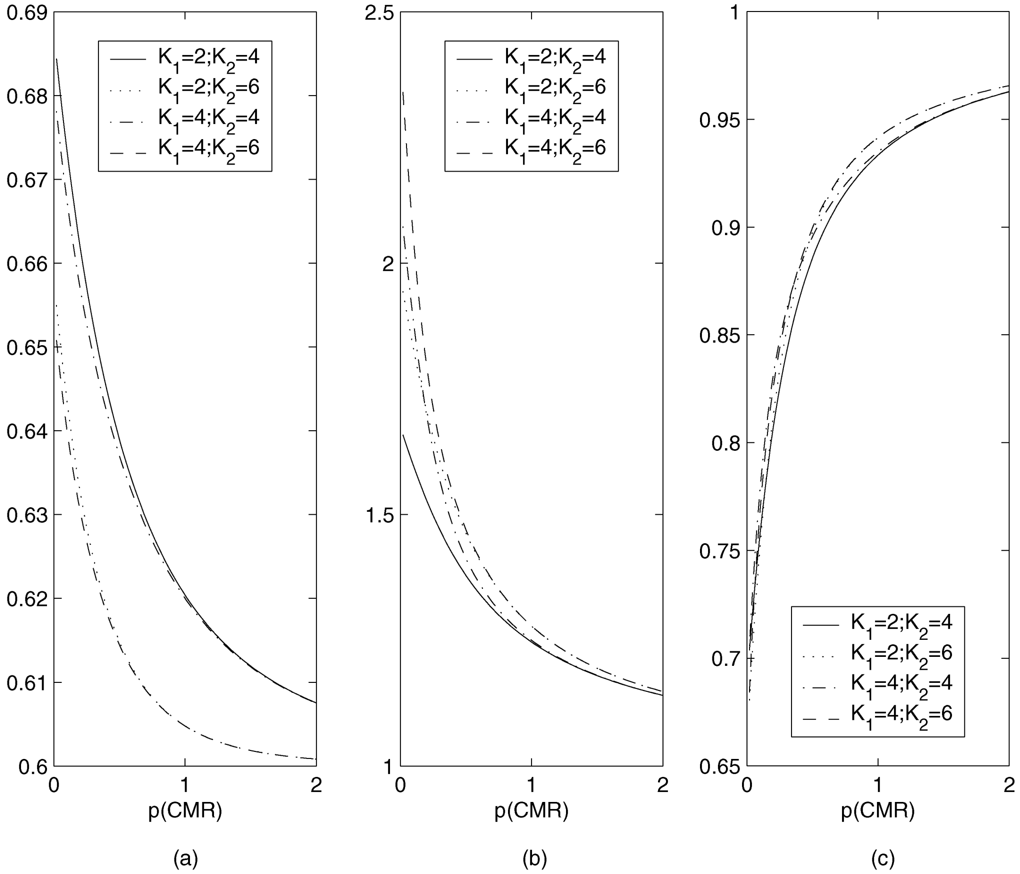


Fig. 6. Relative MOVE and FIND costs of forwarding with $\delta = 0.6$ and $K = 1.5$. (a) The MOVE cost M'/M. (b) The FIND cost F'/F. (c) The net cost C_F/C_B.

traverses more level_1 pointers in the first case. However, there are more local signaling message exchanges in the first case because the level_2 pointer chain threshold is shorter. This is the trade off the operator can make for different classes of users with different QoS requirements.

In the local anchor scheme, a VLR near the user is selected as the local anchor and the user will update his/her location to the local anchor upon every move. The local anchor will not change until a call arrives to the user. The advantage is that the local anchor is usually closer to the user than HLR is, so the total cost will be saved. However, if the user keeps moving away from his/her LA, the cost of updating the location to the LA will become higher and higher and the total cost will also become higher. For the comparison purpose, we will use the same notation in [3] described as follows:

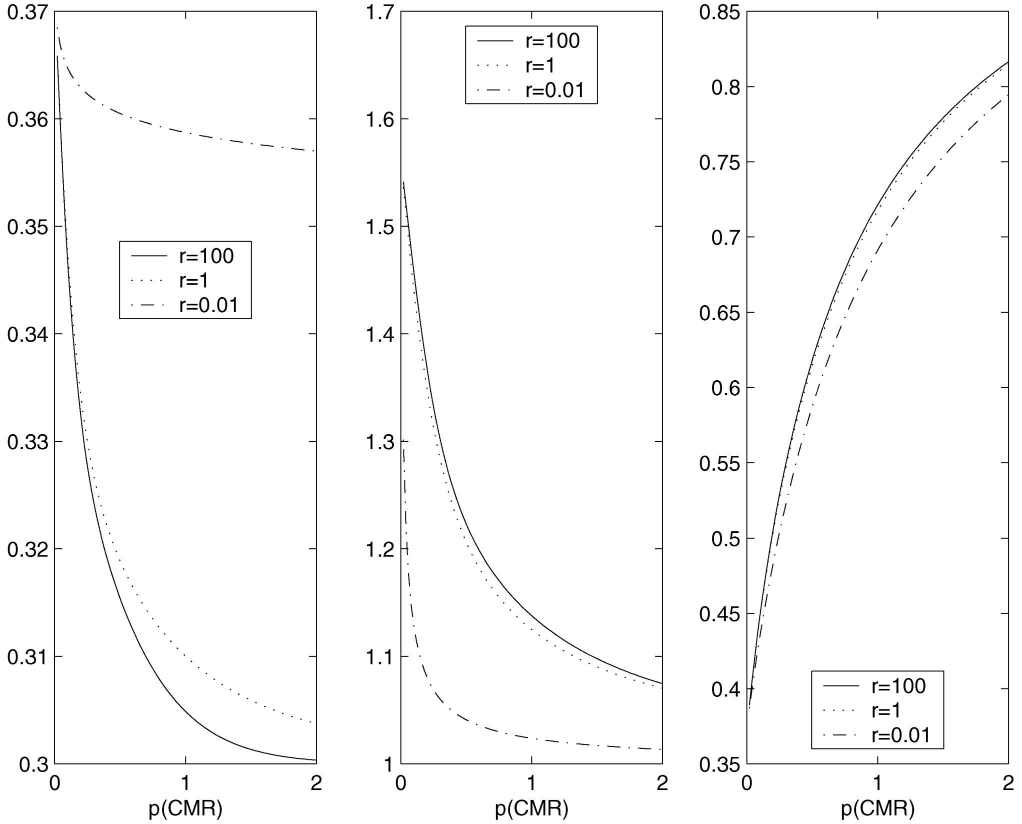
- C_h: Cost for a query or an update of the HLR.
- C_v: Cost for a query or an update of the VLR.
- C_l: Cost for routing a message by the LSTP.
- C_r: Cost for routing a message by the RSTP.
- C_{la}: Cost for sending a signaling message through the local A-link.
- C_{ra}: Cost for sending a signaling message through the remote A-link.
- C_d: Cost for sending a signaling message through the D-link.

All the above parameters can be considered in terms of processing or transmission delay. The results with less value indicate that the time in processing the location update and call delivery costs less. The following notations are also used in [3]:

- h_1: The cost for sending a signaling message from one MSC to another MSC through the HLR, which is given by
$$h_1 = C_h + C_v + 2C_r + 2C_l + 2C_{ra} + 2C_d + 2C_{la}.$$
- h_2: The cost for sending a signaling message from one MSC to another MSC through the LSTP, which is given by $h_2 = C_v + C_l + 2C_{la}$.
- h_3: The cost for sending a signaling message from one MSC to another MSC through the RSTP, which is given by
$$h_3 = C_v + C_r + 2C_l + 2C_d + 2C_{la}.$$

Because the cost in local anchor scheme for location update and call delivery heavily depends on the user location, we need to consider various location scenarios. The following three location types are defined in [3]:

- *HOME*: The mobile is located at the LA.
- *LOCAL*: The mobile is located at an RA other than the LA in the anchor LSTP region.



(a)

(b)

(c)

Fig. 7. The effect of variance in residence time (γ) with $\delta = 0.3$, $K = 1.5$, $K_1 = 4$, and $K_2 = 4$. (a) The MOVE costs M'/M. (b) The FIND cost F'/F. (c) The net cost C_F/C_B.

- *REMOTE*: The mobile terminal is outside of the anchor LSTP region.

The cost of the local anchor scheme is determined by the user's moving direction. If the user keeps moving away from his/her LA before two consecutive calls, the cost will be high; otherwise, the cost is going to be low. The authors in [3] gave nine possible combinations of the location types when an additional movement, the $(n+1)$th movement, is performed after the nth movement (see [3] for details). In this comparison, for simplicity, we choose two scenarios: 1) three movement combination types—the user moves around the LA and the probability for each type is $\frac{1}{3}$ and 2) four movement combination types—the user moves away from the LA, the probability for one type is 0.1 and the probability for each of the rest three types is 0.3. Based on these two scenarios, we compare the costs for the proposed scheme and the local anchor scheme.

According to above definition, the costs for two-level pointer forwarding scheme are given by:

$$\begin{aligned} S_2 &= 2qh_2 + 2(1-q)h_3, \\ T_2 &= \frac{S_2}{2} - 2C_v, \\ m &= 2h_1 - C_h - C_v, \\ F &= 2h_1 - C_h - C_v, \end{aligned}$$

where q is the probability that the user is within its current LSTP for any movement given by the user.

Table 1 shows the relative cost of the LA and two-level pointer forwarding schemes. Here, we assume that the size of an LSTP region is 8×8, so $q = 0.875$ [4]. We also assume that $T_1 = 1.5T_2$, $S_1 = 1.5S_2$, $K_1 = 2$, $K_2 = 4$, and $C_v = C_h = 0.5$. The LA_1 is the case that a user always moves around the LA and never goes far between two consecutive calls (scenario 1). On the contrary, LA_2 is the situation that a user will leave the LA during the call interval with high probability (scenario 2). It can be expected that the values for LA_1 will be smaller than those for LA_2. As we can see from Table 1, h_2 is normalized to one because it is the smallest one. The two-level pointer forwarding strategy could be adopted if its values are smaller than LA_2. For the parameter sets 1, 2, and 4, the two-level pointer forwarding scheme performs better than the LA scheme. The smaller the local signaling cost relative to the long distance signaling cost, the better the result. For the parameter set 3, the cost for sending a signaling message through RSTP is not more expensive than h_2. In this case, it is more efficient to set a long pointer from LA to the user than to set a chain consisting of shorter pointers, which is why the LA_2 is smaller. The parameter set 5 is just for demonstration purpose, it represents an extreme situation such that the cost for sending a message between two MSCs is constant regardless of the path selected. The relative cost for two-level pointer forwarding scheme is larger than 1, which is obvious because the cost of sending a message

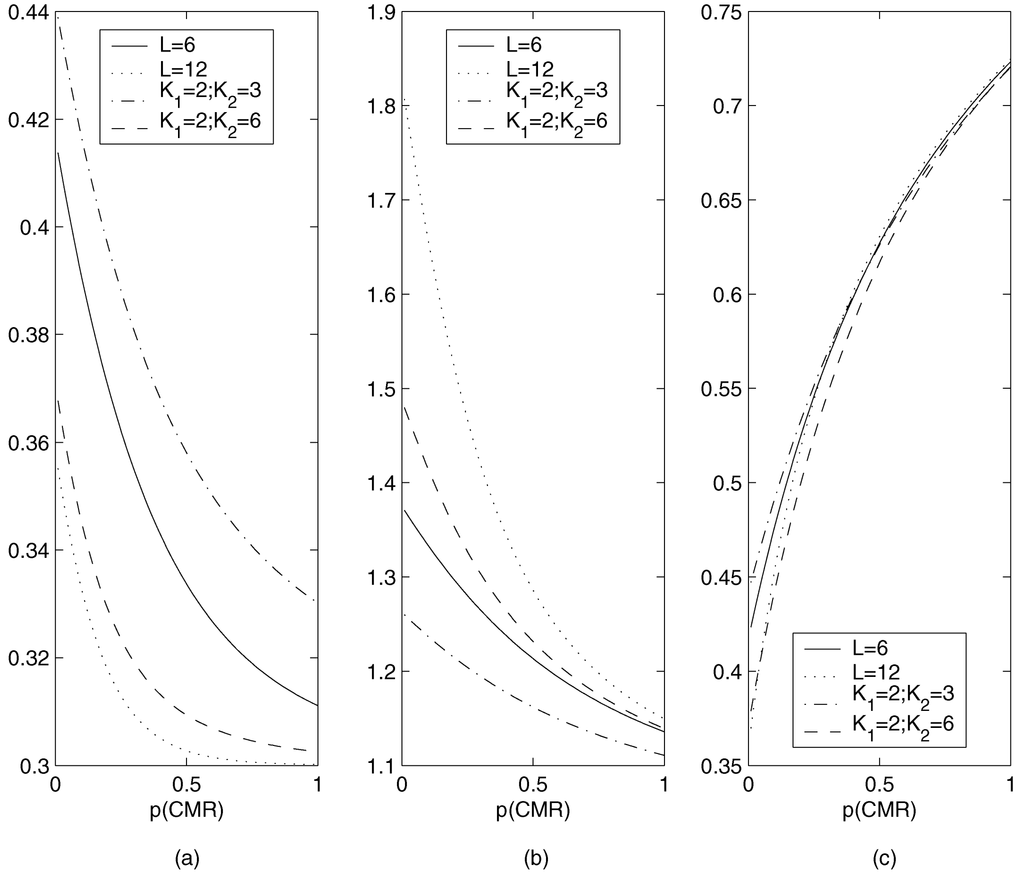


Fig. 8. Relative costs for two-level forwarding and per-user forwarding. (a) The MOVE cost M'/M. (b) The FIND cost F'/F. (c) The net cost C_F/C_B.

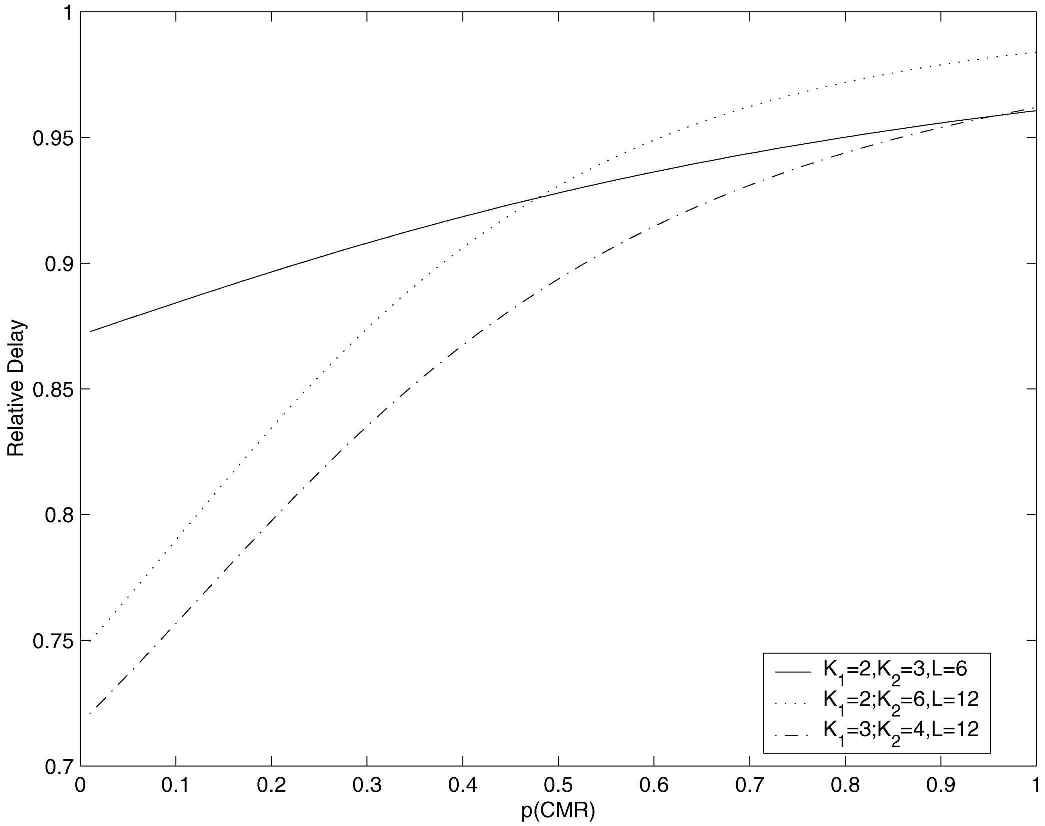


Fig. 9. Relative delay with $\delta = 0.3$ and $K = 1.5$.

TABLE 1
Relative Cost for LA and Two-Level Pointer Forwarding Strategies

Set	h_1	h_2	h_3	LA_1	LA_2	Two_level
1	10	1	9	0.0667	0.615	0.3668
2	10	1	5	0.0667	0.355	0.3153
3	10	1	2	0.0667	0.16	0.2767
4	3	1	3	0.2222	0.75	0.6233
5	1	1	1	0.6667	0.95	1.946

in this case is the same no matter where the user is, hence, it is cheaper to update the location to the HLR directly every time the user moves.

6 Discussions

In the two-level pointer forwarding procedures, the pointer chain can be shortened by the MAs a user traverses. Thus, the updates to the HLR is mitigated at the expense of increase in the local signaling traffic. It is beneficial when the cost of communicating to HLR is relatively higher than the cost of local signaling traffic. One of the advantages of the two-level forwarding strategy is that it can keep the *FIND* penalty low while significantly reducing the total system cost at the same time. As we can see from the previous sections, the per-user forwarding scheme [4] is a special case of the two-level pointer forwarding scheme. When we set $K_1 = 1$ or $K_2 = 1$, the two-level pointer forwarding strategy reduces to the per-user forwarding scheme. There are some other ways to set up the pointer chain. For example, all the PCS service area can be divided into Mobility Regions (MRs). A user will update his/her location to his/her MA in that region. Only when the user moves out of the region is a pointer set up from the old MA to the new one. For the 3G wireless communication systems, 3GPP 23.119 specification proposed an approach to limit the signaling traffic between the visited mobile system and the home mobile system [8], [9]. A new entity *gateway location register* (GLR) is introduced between the VLR/SGSN and the HLR. From the viewpoint of the VLR/SGSN at the visited network, the GLR is treated as the roaming user's HLR located at the home network. From the viewpoint of the HLR at the home network, the GLR acts as the VLR/SGSN at the visited network. Indeed, in the 3G wireless systems, a new level of location management database is added. The users need to exchange extra local messages but reduce the long distance or international messages exchanged in each of the subsequent registrations. The two-level pointer forwarding strategy can be implemented in the 3G systems in the following manner: The GLRs can be selected as the MAs. A realistic implementation of the two-level pointer forwarding scheme should also take into account the possibility that loops may form as the user visits several RAs in succession. Thus, if a user revisits an RA and a pointer for that user is found in that VLR, then the old pointer can be deleted to avoid unnecessary operations. This is called "implicit pointer compression" in [4].

7 Conclusions

In this paper, we propose a new location management scheme, called the two-level pointer forwarding strategy, which intends to reduce the cost of location management by localizing or distributing the signaling traffic and to overcome the HLR bottleneck problem while reducing the call setup (finding) delay. The performance analysis is carried out to show the advantages of the newly proposed scheme. Comparison studies with the per-user pointer forwarding scheme and the local anchor scheme are also undertaken and show that the proposed scheme outperforms either one of the aforementioned schemes. More importantly, the proposed scheme incorporates more parameters to be used to optimize the performance of the location management scheme. Moreover, the proposed scheme can be easily tailored for the 3G wireless systems in which gateway location register is used.

Appendix

In this appendix, we present the derivation of (6) and (7) with the assumptions given in Section 4. We observe that

$$\begin{aligned} M' &= \sum_{i=0}^{\infty}\left\{\left\lfloor\frac{i}{K_1K_2}\right\rfloor m + \left(\left\lfloor\frac{i}{K_2}\right\rfloor - \left\lfloor\frac{i}{K_1K_2}\right\rfloor\right)S_1 \right.\\ &\qquad \left. + \left(i-\left\lfloor\frac{i}{K_2}\right\rfloor\right)S_2\right\}\alpha(i)\\ &= \underbrace{\sum_{i=0}^{\infty} iS_2\alpha(i)}_{X} + \underbrace{\sum_{i=0}^{\infty}\left\lfloor\frac{i}{K_2}\right\rfloor(S_1-S_2)\alpha(i)}_{Y} \qquad (16)\\ &\qquad + \underbrace{\sum_{i=0}^{\infty}\left\lfloor\frac{i}{K_1K_2}\right\rfloor(m-S_1)\alpha(i).}_{Z} \end{aligned}$$

X can be simplified from the definition of $\alpha(i)$,

$$X = S_2\sum_{i=0}^{\infty} i\alpha(i) = \frac{S_2}{p}. \tag{17}$$

The probability $\alpha(i)$ can be expressed as (see [5] for the detailed derivation),

$$\alpha(i) = \frac{(1-g)^2 g^{i-1}}{p}. \tag{18}$$

$$Y = (S_1 - S_2)\sum_{i=0}^{\infty}\left\lfloor\frac{i}{K_2}\right\rfloor\alpha(i). \tag{19}$$

Let $i = jK_2 + k$, then

$$\alpha(jK_2 + k) = \frac{(1-g)^2}{pg}(g^{K_2})^j g^k = yz^jx^k, \tag{20}$$

where

$$y = \frac{(1-g)^2}{pg}, \qquad z = g^{K_2}, \qquad x = g.$$

We can rewrite,

$$\begin{aligned} Y &= y(S_1 - S_2)\sum_{j=0}^{\infty}\sum_{k=0}^{K_2-1} jz^jx^k \\ &= \frac{y(S_1 - S_2)(1 - x^{K_2})}{1 - x}\left(\sum_{j=0}^{\infty} jz^j\right) \\ &= \frac{yz(S_1 - S_2)(1 - x^{K_2})}{(1 - x)(1 - z)^2} \\ &= \frac{(1-g)(S_1 - S_2)g^{K_2-1}}{p(1 - g^{K_2})}. \end{aligned} \tag{21}$$

Similarly,

$$\begin{aligned} Z &= (m - S_1)\sum_{i=0}^{\infty}\left\lfloor \frac{i}{K_1K_2} \right\rfloor \alpha(i) \\ &= \frac{(1-g)(m - S_1)g^{K_1K_2-1}}{p(1 - g^{K_1K_2})}. \end{aligned} \tag{22}$$

So, we obtain

$$M' = \frac{S_2}{p} + \frac{(1-g)g^{K_2-1}(S_1 - S_2)}{p(1 - g^{K_2})} + \frac{(1-g)g^{K_1K_2-1}(m - S_1)}{p(1 - g^{K_1K_2})}. \tag{23}$$

We can derive F' in a similar way,

$$\begin{aligned} F' &= F + \sum_{i=0}^{\infty}\Bigg\{\left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor T_1 \\ &\quad + \left(i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2 - \left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor K_2\right)T_2\Bigg\}\alpha(i) \\ &= F + \underbrace{(T_1 - K_2T_2)\sum_{i=0}^{\infty}\left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor \alpha(i)}_{U} \\ &\quad + \underbrace{T_2\sum_{i=0}^{\infty}\left(i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2\right)\alpha(i).}_{V} \end{aligned} \tag{24}$$

Let $i = jK_1K_2 + k$, then

$$\alpha(jK_1K_2 + k) = \frac{(1-g)^2}{pg}(g^{K_1K_2})^j g^k = yz^jx^k, \tag{25}$$

where

$$y = \frac{(1-g)^2}{pg}, \qquad z = g^{K_1K_2}, \qquad x = g. \tag{25}$$

$$\begin{aligned} U &= (T_1 - K_2T_2)\sum_{i=0}^{\infty}\left\lfloor \frac{i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2}{K_2} \right\rfloor \alpha(i) \\ &= (T_1 - K_2T_2)\sum_{j=0}^{\infty}\sum_{k=0}^{K_1K_2-1}\left\lfloor \frac{k}{K_2} \right\rfloor \alpha(jK_1K_2 + k) \\ &= (T_1 - K_2T_2)\sum_{j=0}^{\infty}\sum_{k=0}^{K_1K_2-1}\left\lfloor \frac{k}{K_2} \right\rfloor yz^jx^k \\ &= (T_1 - K_2T_2)y\sum_{j=0}^{\infty} z^j \sum_{k=0}^{K_1K_2-1}\left\lfloor \frac{k}{K_2} \right\rfloor x^k \\ &= (T_1 - K_2T_2)y\sum_{j=0}^{\infty} z^j \left(\sum_{n=0}^{K_1-1}\sum_{m=0}^{K_2-1} nx^{nK_2+m}\right) \\ &= \frac{(T_1 - K_2T_2)(1-g)\left[g^{K_2} - K_1g^{K_1K_2} + (K_1 - 1)g^{(K_1+1)K_2}\right]}{pg(1 - g^{K_1K_2})(1 - g^{K_2})}. \end{aligned} \tag{26}$$

$$\begin{aligned} V &= T_2\sum_{i=0}^{\infty}\left(i - \left\lfloor \frac{i}{K_1K_2} \right\rfloor K_1K_2\right)\alpha(i) \\ &= T_2\sum_{j=0}^{\infty}\sum_{k=0}^{K_1K_2-1} k\alpha(jK_1K_2 + k) \\ &= yT_2\sum_{j=0}^{\infty} z^j \sum_{k=0}^{K_1K_2-1} kx^k \\ &= \frac{\left[1 - K_1K_2g^{K_1K_2-1} + (K_1K_2 - 1)g^{K_1K_2}\right]T_2}{p(1 - g^{K_1K_2})}. \end{aligned} \tag{27}$$

So we have

$$\begin{aligned} F' &= F + \frac{T_2\left[1 - K_1K_2g^{K_1K_2-1} + (K_1K_2 - 1)g^{K_1K_2}\right]}{p(1 - g^{K_1K_2})} \\ &\quad + \frac{(T_1 - K_2T_2)(1-g)\left[g^{K_2} - K_1g^{K_1K_2} + (K_1 - 1)g^{(K_1+1)K_2}\right]}{pg(1 - g^{K_1K_2})(1 - g^{K_2})}. \end{aligned} \tag{28}$$

ACKNOWLEDGMENTS

The authors would like to express their gratitude to the reviewers for their comments which greatly enhanced the quality of this paper. This work was supported in part by the US National Science Foundation Faculty Early Career Development Award ANI-0093241.

REFERENCES

[1] EIT/TIA, "Cellular Radio Telecommunications Intersystem Operations,"Technical Report IS-41 (Revision B), EIA/TIA, 1991.

[2] ETSI, *Digital Cellular Telecommunications System (Phase 2+): Mobile Application Part (MAP) Specification (GSM 09.02 version 7.51 Release).* 1998.

[3] J. Ho and F. Akyildiz, "Local Anchor Scheme for Reducing Signaling Costs in Personal Communications Networks," *IEEE/ACM Trans. Networking,* vol. 4, no. 5, Oct. 1996.

[4] R. Jain and Y.B. Lin, "An Auxiliary User Location Strategy Employing Forwarding Pointers to Reduce Network Impacts of PCS," *Wireless Networks 1,* pp. 197-210, 1995.

[5] Y. Fang, I. Chlamtac, and Y.B. Lin, "Portable Movement Modeling for PCS Networks," *IEEE Trans. Vehicular Technology,* vol. 87, no. 8, pp. 1347-1384, Aug. 1999.

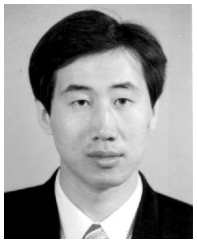
[6] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu, and W. Wang, "Mobility Management in Next-Generation Wireless Systems," *Proc. IEEE,* vol. 4, no. 5, Oct. 1996.

[7] S. Tabbane, "Location Management Methods for Third-Generation Mobile System," *IEEE Comm. Magazine,* Aug. 1997.

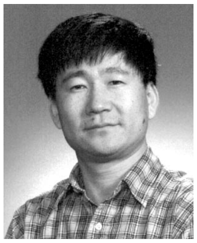
[8] Y.B. Lin and I. Chlamtac, *Wireless and Mobile Network Architectures.* John Wiley & Sons, Inc, 2001.

[9] Y.B. Lin, Y.R. Haung, Y.K. Chen, and I. Chlamtac, "Mobility Management: from GPRS to UMTS," *Wireless Comm. and Mobile Computing (WCMC),* vol. 1, no. 4, pp. 339-359, 2001.

[10] R. Jain, Y.B. Lin, C. Lo, and S. Mohan, "A Caching Strategy to Reduce Network Impacts of PCS," *IEEE J. Selected Areas in Comm.,* vol. 12, no. 8, Oct. 1994.



Wenchao Ma (S'99) received the BS and MS degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 1995 and 1998, respectively. From 1998 to 1999, he was an engineer with China Telecom. From 1999 to 2000, he was a PhD student in the Electrical and Computer Engineering Department at New Jersey Institute of Technology. He is currently pursuing the PhD degree in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, where he is a research assistant in the Wireless Networks Laboratory (WINET). His research interests include wireless multimedia networks, mobility management, mobile computing, and IP mobility management. He is a student member of the IEEE.



Yuguang Fang (S'92-M'94-S'96-M'97-SM'99) received the BS and MS degrees in mathematics from Qufu Normal University, Qufu, Shandong, China, in 1984 and 1987, respectively, a PhD degree in systems and control engineering from the Department of Systems, Control, and Industrial Engineering at Case Western Reserve University, Cleveland, Ohio, in January 1994, and a PhD degree in electrical engineering from the Department of Electrical and Computer Engineering at Boston University, Massachusetts, in May 1997. From 1987 to 1988, he held research and teaching positions in both the Department of Mathematics and the Institute of Automation at Qufu Normal University. He held a postdoctoral position in the Department of Electrical and Computer Engineering at Boston University from June 1994 to August 1995. From June 1997 to July 1998, he was a visiting assistant professor in the Department of Electrical Engineering at the University of Texas at Dallas. From July 1998 to May 2000, he was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology, Newark. Since May 2000, he has been an assistant professor in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville. His research interests span many areas including wireless networks, mobile computing, mobile communications, automatic control, and neural networks. He has published more than 60 papers in refereed professional journals and conferences. He has received the US National Science Foundation Faculty Early Career Award in 2001 and the US Office of Naval Research Young Investigator Award in 2002. He is listed in Marquis *Who's Who in Science and Engineering*, *Who's Who in America*, and *Who's Who in the World*. Dr. Fang has actively engaged in many professional activities. He is a senior member of the IEEE and a member of the ACM. He is an editor for *IEEE Transactions on Communications*, an editor for *IEEE Transactions on Wireless Communications*, an editor for *ACM Wireless Networks*, an area editor for *ACM Mobile Computing and Communications Review*, an associate editor for *Wiley International Journal on Wireless Communications and Mobile Computing*, and feature editor for Scanning the Literature in *IEEE Personal Communications*. He was an editor for the *IEEE Journal on Selected Areas in Communications: Wireless Communications Series*. He has also been actively involved with many professional conferences such as ACM MobiCom'02, ACM MobiCom'01, IEEE INFOCOM'00, INFOCOM'98, IEEE WCNC'02, WCNC'00 (Technical Program Vice-Chair), WCNC'99, and the International Conference on Computer Communications and Networking (IC3N'98) (Technical Program Vice-Chair).