

# One-Step Pointer Forwarding Strategy for Location Tracking in Distributed HLR Environment

Kuen-Liang Sue and Chien-Chao Tseng

**Abstract**—One of the main challenges in personal communication service (PCS) is to locate many mobiles that may move frequently from place to place. Such a system operation is called location tracking. Many network signals flow, and database queries are required to achieve such a task. In addition to the two-level hierarchical strategy in IS-41 and GSM, several strategies have been proposed to improve the efficiency of location tracking. Pointer forwarding was used to reduce the expensive home location register (HLR) accesses. Previously, the distributed HLR scheme was proposed to prevent HLR from becoming a bottleneck in the signaling network. However, the length of a forwarding pointer chain may be lengthened in a distributed HLR environment. In this paper, we propose a more efficient strategy to overcome this potential problem. This strategy attempts to migrate the locating chains in a distributed HLR system when a mobile issues a registration operation. As a consequence, the length of any forwarding pointer chain does not exceed one in our strategy. Simulation results indicate that our strategy significantly decreases the locating cost. In fact, this strategy provides an upper bound of location tracking time owing to the fact that the length of any locating path does not exceed one. Furthermore, obsolete entries in local databases visitor location registers can be reclaimed in this strategy.

In addition to a two-level hierarchical strategy in IS-41 and GSM [2], several methods have been proposed to improve the efficiency of location tracking [3]–[8], [10]. The pointer forwarding strategy has been proposed to avoid the expensive HLR access each time a mobile moves to a new registration area (RA) [7]. Lin suggested that distributing HLR's in several areas can prevent HLR from becoming a signal bottleneck in the signaling network [8]. However, in the IS-41 and GSM protocols, distributed HLR requires multiple HLR updates to maintain all distributed HLR's containing the valid locating information. In that study, the forwarding pointer strategy and the concept of distributed HLR's are combined to provide a new locating tracking strategy. It uses the characteristic of forwarding pointers to make the concept of distributed HLR [3] more attractive and efficient. However, a potential limitation of the strategy is that the length of the forwarding pointer chains may become too long. Therefore, this paper provides a more efficient strategy to overcome such a limitation, and makes the notion of distributed HLR's more feasible.

## I. INTRODUCTION

THE personal communication service (PCS) is a system that aims to allow for communication anywhere in the world. Such a system attempts to locate many mobiles that may move frequently from place to place [1]. Location tracking operation in a PCS network is expensive because that many signal flow and database queries are needed to achieve such a task. Moreover, locating a mobile is also a time-consuming process. In fact, the time to deliver an incoming call to a mobile is dominated by the locating time. If a location tracking strategy is efficient, the call delivery time is shortened significantly. Therefore, devising a good location tracking strategy is necessary in PCS systems.

Common location tracking strategies use a two-level hierarchical mechanism to maintain two classes of databases of user location information [2]. One class of the databases is called the home location register (HLR), and the other is the visitor location register (VLR).

Manuscript received September 12, 1996; revised April 16, 1997. This work was supported in part by the National Science Council, R.O.C., under Grant NSC86-2213-E-009-076.

K.-L. Sue is with the Department of Information Management, Ling Tung College, Taichung, Taiwan, R.O.C.

C.-C. Tseng is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier: S 0733-8716(97)06087-3.

## II. RELATED LOCATION TRACKING STRATEGIES

### A. Location Tracking in IS-41

In the IS-41 protocol, all service areas are divided into many registration areas (RA's) [12]. To record every mobile's location information, there are two kinds of databases, HLR/VLR. When a user subscribes to the service, a record associated with this user is created in the system database, HLR. As a mobile roams and arrives at a new RA, a record for this mobile is created in this RA's database, i.e., VLR. In fact, some RA's may share a VLR. To simplify research, we assume that every RA has its own VLR. Notably, the mobile switching center (MSC) is near the associated VLR [2]. "VLR" is used to represent "MSC/VLR" later. Signaling flow between MSC and VLR will be ignored in this paper.

In IS-41, if a mobile moves from some RA to another one, it must be registered at the VLR of the new RA by sending a registration-request message [2]. This VLR creates a temporary record for the mobile, and sends a message to inform the HLR about the mobile's new location. To erase the obsolete record in the VLR of the old RA, HLR sends a registration-cancel message to the old VLR [2], [9], [11]. The composition of the above actions is referred to as *registration operation*, and is shown in Fig. 1.

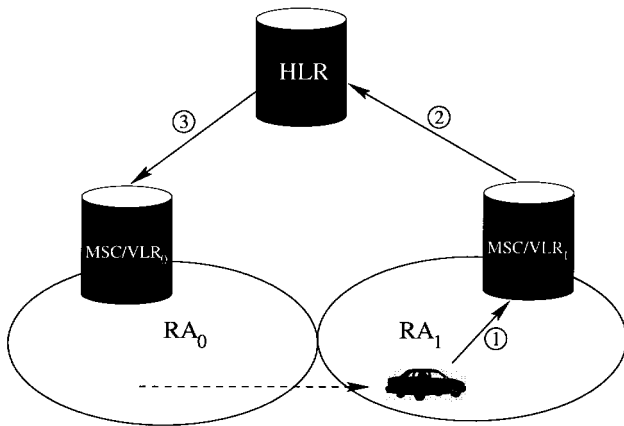


Fig. 1. Registration in IS-41.

When the PCS system attempts to deliver a call to a mobile, another procedure called *call delivery* is required. Location tracking is the primary task in this procedure as described below [2], [13].

- 1) When an incoming call occurs, a table lookup technique called global title translation (GTT) is required at the signal transfer pointer (STP) to identify the address of the HLR serving the called mobile  $m$ .
- 2) A *location-request* message is sent to query the HLR of the mobile  $m$ .
- 3) The HLR determines the VLR currently serving  $m$ , and queries it by sending a *route-request* signal.
- 4) The VLR forward the query message to the mobile switching center (MSC). If the mobile  $m$  can receive the call, the MSC returns a routable address called the temporary local directory number (TLDN) to the VLR.
- 5) The VLR forward the TLDN back to the originating MSC via the HLR of the mobile  $m$ .
- 6) When the originating MSC receives the TLDN, it routes the call to the MSC where the mobile  $m$  is located.

By cooperating with these two procedures, the system can locate a mobile and deliver a call to it correctly [2].

### B. Pointer Forwarding with Single HLR Scheme

For a mobile which frequently moves across RA's but seldom has an incoming call, traditional IS-41 registration operation is wasted in the network cost. In traditional IS-41, every registration operation must update HLR to record the most recent location of a mobile. However, never receiving any call, the mobile may leave for another RA again, and initiate another registration operation in the new RA. In fact, access to HLR is much more expensive than access to VLR. Based on these observations, the pointer forwarding scheme is proposed to reduce the waste.

In a pointer forwarding scheme, registration operation is changed slightly [6], [7]. A mobile coming to a new RA sends a registration message to the VLR of the new RA. Instead of sending a message to inform HLR, the new VLR sends a message to inform the old VLR about the mobile's leaving. On receiving this message, the old VLR deletes the obsolete record

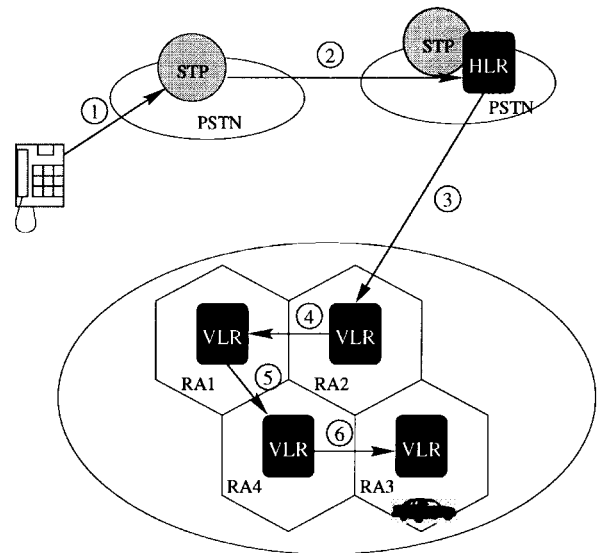


Fig. 2. Location tracking in PFSHLR.

for the mobile, and creates a forwarding pointer pointing to the new RA [6]. Fig. 2 depicts the state transition, and also shows the process of locating a mobile.

### C. The Pointer Forwarding with Distributed HLR's Scheme

As described in Section II-A, whenever a call occurs in a single HLR environment, a locating-request message must be sent to the only HLR. The HLR must determine the VLR currently serving the callee and query it by sending a route-request signal. Therefore, the HLR may become a bottleneck due to the heavy signaling traffic generated by location tracking. Distributing HLR's in several regions is a highly effective method of dispersing the traffic load in HLR. However, in IS-41, distributed HLR's require multiple HLR updates when a mobile registers at a new RA. Lin proposed a pointer forwarding with a distributed HLR's (PFDHLR) scheme to eliminate this overhead in a distributed HLR environment [8].

In this scheme, HLR's are distributed in remote public switching telephone networks (PSTN's). PFDHLR assumes that a distributed HLR is near the STP that performs GTT. Like pointer forwarding with a single HLR (PFSHLR) strategy, every HLR in PFDHLR records an RA where a mobile may exist. In fact, the RA is the position where the mobile is found the last time by individual HLR's. Different distributed HLR's may imply different RA's. All registration processes are the same as in PFSHLR. That is, merely building up a pointer from the old VLR to the new VLR without access to any HLR is used in this scheme.

When an incoming call is originated from some remote PSTN, the HLR will be directly queried. This fact suggests that GTT is not required because the distributed HLR is near STP. This is another benefit resulting from a distributed HLR environment. Next, the HLR at that PSTN queries the VLR recorded in the entry associated with the callee. If the mobile is found in this RA, a call will be delivered to it. Otherwise, the

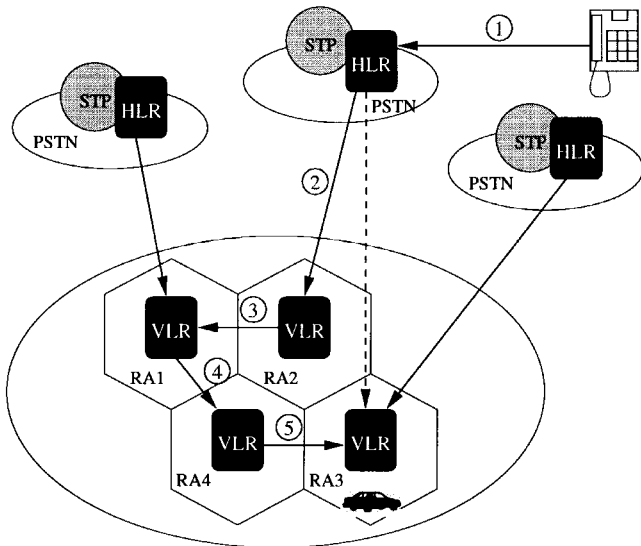


Fig. 3. Location tracking in PFDHLR.

system will trace the mobile through the forwarding pointer chain starting from that RA. Fig. 3 depicts this call delivery process. After completing the locating operation, the HLR will change the record for the called mobile to point to the RA where the mobile is found, as Fig. 3 indicates by the dotted line.

This scheme effectively and efficiently implements the distributed HLR notion. However, it has some potential side effects. The length of the pointer chain may be lengthened. Because of the distributed HLR’s in several regions, the total incoming calls will be shared by all HLR’s. Therefore, the average call arrival rate in each distributed HLR is  $1/N$  times that in PFSHLR. The call arrival rate divided by the frequency of crossing the RA boundary is called the call-to-mobility ratio. Restated, the call-to-mobility ratio for each distributed HLR is  $1/N$  times that in PFSHLR. According to the simulation of [8], the expected length of each pointer chain increases considerably when the call-to-mobility ratio is small.

### III. LOCATION TRACKING WITH ONE-STEP POINTER FORWARDING AND DISTRIBUTED HLR

To overcome the potential problem in PFDHLR, we propose a new location tracking strategy with one-step pointer forwarding and distributed HLR’s.

#### A. Overview of Our Strategy

Similar to PFDHLR, our strategy also incorporates the concept of distributed HLR’s with a pointer forwarding. Every distributed HLR has one record for an authorized mobile. This records the individual RA/VLR where a mobile is found the last time by an individual HLR. Different HLR’s may record different RA/VLR’s as the head of their own locating paths, so they may have different locating paths; therefore, one cellular system with  $N$  distributed HLR’s may have  $n$  different locating pointer chains,  $n \leq N$ , for the same mobile. Whenever a mobile registers at a new RA, the VLR in the

Prev.			TAG			Mobile_ID	Next
0	1	2	0	1	2	Integer	P
P	P	P	B	B	B		

P : Pointer      B : Bit

Fig. 4. Data structure of each entry in VLR.

new RA will maintain an entry for this mobile. This entry is the common rear of the  $n$  locating pointer chains. Similar to a double link list, every such entry must record the previous nodes in the  $n$  locating pointer chains. In this manner, VLR can send a deregistration message through all locating paths. This message is forwarded to proper VLR’s via the previous link in each entry. Next, all locating paths are migrated to guarantee that each pointer chain’s length among VLR’s does not exceed one. This property is always true regardless of from which HLR an incoming call originates. Based on this characteristic, we call it the “one-step pointer forwarding strategy for location tracking in a distributed HLR environment.”

#### B. Data Structure Definition and Assumption

Before describing how our strategy works, the related data structure must be defined. For simplicity, we assume that there are three distributed HLR’s in a cellular system. Fig. 4 represents the structure of an entry. The four fields are explained individually as follows.

- 1) Prev: This is a pointer array. “Prev” is an abbreviation for “previous.” In this example, there are three elements in this array. Each element is the address of a previous VLR in some locating pointer chain. This system may have three locating chains existing at the same time since there are three distributed HLR’s in the system and each HLR has its own searching path. Thus, a VLR can appear in three paths at most.
- 2) TAG: This is a 3-bit flag. Each bit corresponds to one distributed HLR. If any HLR<sub>*i*</sub> records that the mobile associated with this entry is in this VLR now, TAG<sub>*i*</sub> will be set to 1. Otherwise, TAG<sub>*i*</sub> is always 0. This flag will be used to determine whether this entry will be reclaimed. Mobile ID: This field records the ID of the mobile associated with this entry.
- 3) Next: This is a forwarding pointer which points to the next VLR in some locating chains.

#### C. One-Step Pointer Forwarding Strategy

In this section, we describe our one-step pointer forwarding strategy in a distributed HLR environment in terms of three system events: handset initiation, registration, and call delivery. The necessary signals and the corresponding procedures for these events are presented in detail in the following subsections.

1) *Handset Initiation*: When a mobile  $m_0$  is initialized in some RA, it will send an INIT( $m_0$ ) signal to the VLR of this RA. As the VLR receives INIT( $m_0$ ), it must perform the following two actions.

- Create an entry for the mobile  $m_0$ .
- Inform all distributed HLR's of this initiation.

By the above actions, all of the records for  $m_0$  in all distributed HLR's will point to this VLR. Therefore, all distributed HLR's know where to deliver calls to this mobile now. This VLR has become the head node in all of the forwarding pointer chains for  $m_0$ .

2) *Registration*: When a mobile  $m_0$  moves across the boundary of an RA, e.g., from  $RA_0$  to  $RA_1$ , it will send a REGISTRATION( $m_0, V_0$ ) signal to the VLR of the new RA. When the VLR receives the REGISTRATION signal, the following actions must be performed.

- Inform the old VLR of the mobile's leaving by sending a DEREGISTRATION signal.
- Create/modify the entry for this mobile.

Sending a DEREGISTRATION to the old VLR will initiate proper modification in all searching paths to maintain the length of all locating paths not more than one.

On receiving the DEREGISTRATION signal, the old VLR reacts as follows.

- Forward the DEREGISTRATION signal to each VLR in the pointer array Prev.  
In some case [16], the new VLR originating from the DEREGISTRATION signal may exist in the Prev array. To avoid unnecessary signal traffic, the old VLR will not forward the DEREGISTRATION signal to the new VLR.
- If all TAG values of the designated entry are 0, the VLR reclaims the entry.  
 $m_0$  has left, and no locating path for  $m_0$  goes via this VLR, so the entry for  $m_0$  can be reclaimed.
- If any TAG value is not 0, perform the following.
  - a) Set the pointer Next to point to the VLR originating from the DEREGISTRATION signal.
  - b) Clear the pointer array Prev.
  - c) Send the ACK signal to the VLR from which this DEREGISTRATION signal originated.

This case implies that this VLR is still in some locating path, and this entry cannot be reclaimed. Therefore, we set Next to point to the current position where  $m_0$  is now located.

On receiving the forwarded DEREGISTRATION signal, a VLR will perform the same actions as described above.

When the new VLR receives signal ACK, it must do one thing:

- Add the RA which originates this ACK to the previous array.

3) *Call Delivery*: If an HLR  $H_0$  wants to deliver a call to a mobile  $m_0$ , it will send a Call\_Setup\_Req( $m_0, H_0$ ) signal to the VLR pointed to by the entry associated with  $m_0$  in  $H_0$ . When the VLR receives this signal, it determines whether or not the mobile  $m_0$  has left.

If  $m_0$  has left, the VLR will perform the following two actions.

- Forward the Call\_Setup\_Req( $m_0, H_0$ ) signal to the VLR pointed to by Next pointer.
- Reset TAG<sub>0</sub>, and reclaim this entry if all TAG's are 0.

If  $m_0$  still exists, the VLR will return the call setup result to  $H_0$ , and set TAG<sub>0</sub> to 1. On receiving the call setup result,  $H_0$  will change the record associated with  $m_0$  to point to the VLR. The call setup result will be a TLDN if  $m_0$  can receive the call now; otherwise, the call setup result will be a failure message.

4) *The Detailed Protocol*: Our protocol was explained from a high-level perspective in the previous subsection. All of the detailed protocols are represented with pseudocodes as shown in Fig. 5. We have programmed this protocol to check its validity. A detailed validity analysis can be found in [16]. According to those results, this protocol always works well, as expected. It guarantees that not more than one forwarding pointer will be traced to locate a mobile regardless of from which remote PSTN an incoming call originates.

#### D. An Illustrative Example

This section presents an illustrative example which demonstrates how this protocol works when a series of different events occurs. This example is illustrated via a series of graphs to facilitate a thorough understanding of this protocol. For simplicity, it is assumed that there are three distributed HLR's in remote PSTN's and four VLR's in our system. The events occur as follows.

- 1) Mobile  $m_0$  is initialized at  $RA_1$ . The VLR of  $RA_1$ , henceforth referred as VLR<sub>1</sub>, informs all three HLR's in a system to record the current position of  $m_0$ , as Fig. 6 indicates.
- 2) An incoming call occurs from PSTN<sub>2</sub>. After this process, no pointer will be changed.
- 3)  $m_0$  moves from  $RA_1$  to  $RA_4$ , and sends a REGISTRATION message to the VLR of  $RA_4$ , VLR<sub>4</sub>. VLR<sub>4</sub> sends a DEREGISTRATION message to VLR<sub>1</sub>, and a forwarding pointer is built in VLR<sub>1</sub> which points to VLR<sub>4</sub>.
- 4)  $m_0$  moves from  $RA_4$  to  $RA_2$ . After receiving the REGISTRATION message, VLR<sub>2</sub> sends a DEREGISTRATION message to  $RA_4$ . VLR<sub>4</sub> sends this message to VLR<sub>1</sub>. Then, VLR<sub>1</sub> adjusts its forwarding pointer, as Fig. 7 reveals. In our protocol, the entry for  $m_0$  in VLR<sub>4</sub> is invalid now, and is reclaimed.
- 5) When  $m_0$  is in  $RA_2$ , an incoming call originates from PSTN<sub>3</sub> for  $m_0$ . The forwarding pointers are traced to find  $m_0$ , and  $m_0$  is found at  $RA_2$ . After this call, the HLR at PSTN<sub>3</sub> will update its record about  $m_0$ 's position to point to  $RA_2$ .
- 6) Another call for  $m_0$  immediately occurs from PSTN<sub>2</sub>. The process for this call is similar to that for event 5). The corresponding system states are shown in Fig. 8.
- 7) After event 6),  $m_0$  moves back to  $RA_1$  from  $RA_2$ . On receiving a REGISTRATION signal, VLR<sub>1</sub> sends a DEREGISTRATION signal to VLR<sub>2</sub>. The pointer from VLR<sub>1</sub> to VLR<sub>2</sub> is eliminated, and another one is constructed from VLR<sub>2</sub> to VLR<sub>1</sub>.

EVENT	Received Signal	Corresponding Procedure						
Initiation	INIT( $m_0$ )	create a entry <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>*</td><td>*</td><td>*</td><td>1</td><td>1</td><td>1</td></tr></table> $m_0$ * for $m_0$ inform all HLRs about this Initiation	*	*	*	1	1	1
*	*	*	1	1	1			
registration	REGISTRATION( $m_0, V_0$ )	send DEREGISTRATION( $m_0, V_1$ ) to $V_0$ if there is already a entry for $m_0$ Next=nil else create a entry <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>*</td><td>*</td><td>*</td><td>0</td><td>0</td><td>0</td></tr></table> $m_0$ * for $m_0$ endif	*	*	*	0	0	0
	*	*	*	0	0	0		
	DEREGISTRATION( $m_0, V_0$ )	if existing a entry for $m_0$ for each element $V_x$ in previous array if $V_x \neq V_0$ forward DEREGISTRATION( $m_0, V_0$ ) to $V_x$ endif if OR(TAG <sub>0</sub> , TAG <sub>1</sub> , TAG <sub>2</sub> )=0 reclaim this entry else Next= $V_0$ clear previous array send Ack( $m_0, V_1$ ) to $V_0$ endif endif						
Ack( $m_0, V_0$ )	add $V_0$ to the previous array of the entry							
Call delivery	Call_Setup_Req( $m_0, H_0$ )	if $m_0$ has left this RA forward Call_Setup_Req( $m_0, H_0$ ) to Next clear TAG <sub>0</sub> =0 if OR(TAG <sub>0</sub> , TAG <sub>1</sub> , TAG <sub>2</sub> )=0 reclaim this entry else if $m_0$ is busy or none of TLDN return fail( $m_0, V_0$ ) else return sucess( $m_0, TLDN$ ) endif set Tag <sub>0</sub> =1 endif endif						

- 1. OR is bit-based or operation.
- 2. '\*' stands for nil.

Fig. 5. Related signals and procedures for each event in OSPFDHLR.

8)  $m_0$  roams from RA<sub>1</sub> to RA<sub>3</sub>. VLR<sub>3</sub> informs VLR<sub>1</sub> about  $m_0$ 's leaving by sending a DEREGISTRATION signal. VLR<sub>1</sub> forward it to VLR<sub>2</sub>. Then, the forwarding pointers from VLR<sub>1</sub> and VLR<sub>2</sub> to VLR<sub>3</sub> are constructed, respectively. The old pointer from VLR<sub>2</sub> to VLR<sub>1</sub> is cancelled. Fig. 9 summarizes the final results.

#### IV. THE PERFORMANCE ANALYSIS MODEL

This section introduces the analytical model of OSPFDHLR, PFDHLR, and PFSHLR for the call delivery network cost in detail. For fairness and convenience, we follow the performance analysis model in [8] which has made a comparison between PFDHLR and PFSHLR in terms of network cost.

#### A. Assumptions

For both PFDHLR and OSPFDHLR, HLR's are distributed in remote PSTN's. A reasonable assumption is that each distributed HLR is near the STP.

In addition, some network cost estimates are made as follows to simplify the comparison.

- 1) The message-passing cost from the caller to the GTT STP in PFSHLR is normalized to 1.
- 2) In both OSPFDHLR and PFDHLR, the message-passing cost from the caller to the distributed HLR is 1.
- 3) The message-passing cost from the GTT STP to the HLR in PFSHLR is 1.

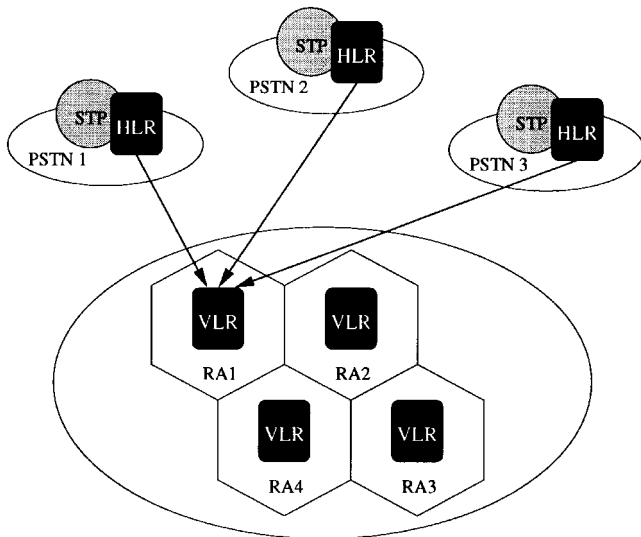


Fig. 6. Searching paths after  $m_0$  is initialized at RA<sub>1</sub>.

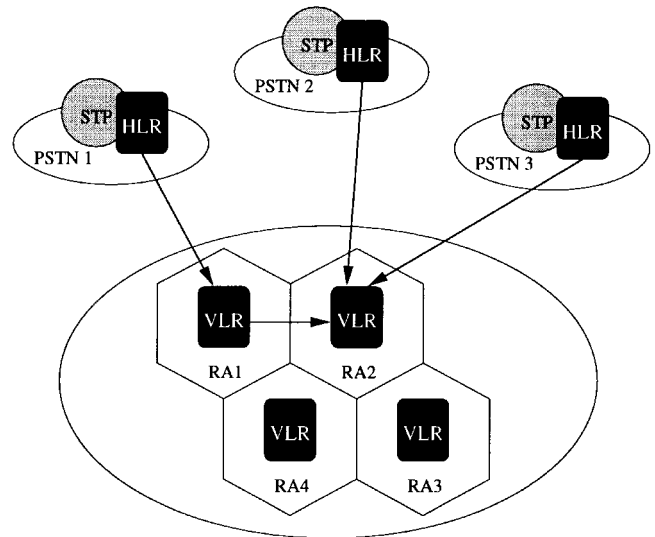


Fig. 8. Searching paths after the call delivery from PSTN<sub>2</sub>.

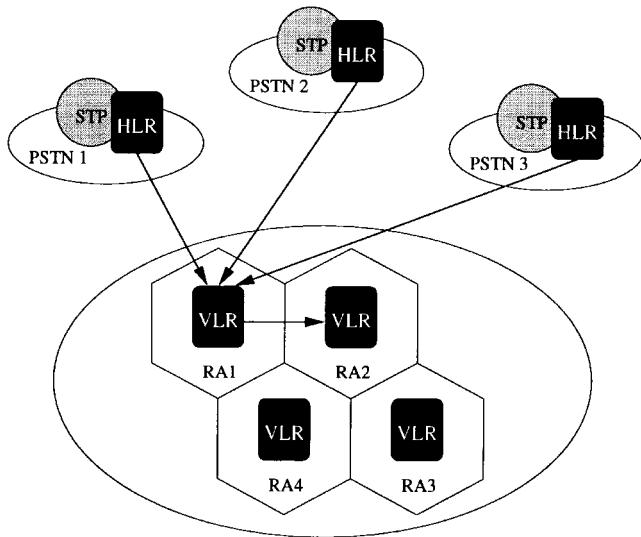


Fig. 7. Searching paths after  $m_0$  moves from RA<sub>4</sub> to RA<sub>2</sub>.

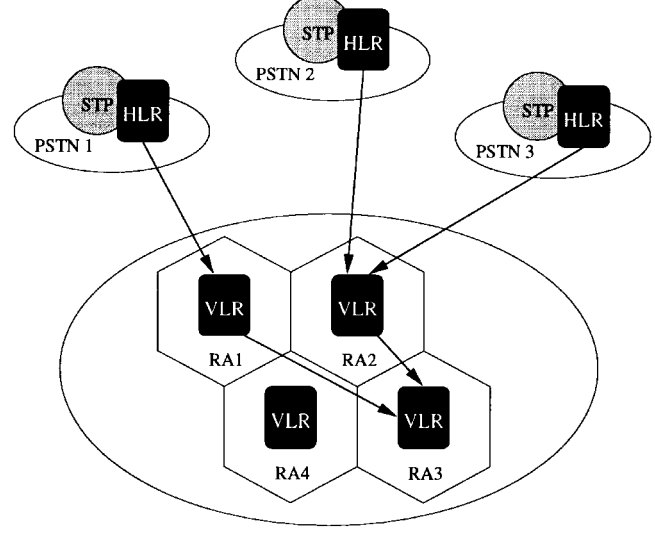


Fig. 9. Searching paths after  $m_0$  moves from RA<sub>1</sub> to RA<sub>3</sub>.

- 4) The query cost from an HLR to a VLR in all three strategies is 1.
- 5) The pointer traversing cost from one VLR to another is  $\delta$ .

The communication cost from the call originator to the STP could be different in different PSTN's, and so is that from the STP to the HLR. However, we would like to focus our research on the effect of shortening the pointer chains among VLR's to reduce the locating cost. Therefore, we normalize these costs to 1, and consider the cost between VLR's as a variable for the sake of simplicity and clarity. Furthermore, the originator-to-STP (or STP-to-HLR) to VLR-to-VLR cost ratio will affect the numerical analysis result. By fixing the cost of originator-to-STP (or STP-to-HLR) and varying the cost of VLR-to-VLR, we have a generalized model to evaluate our scheme applied to different networks with different underlying communication costs.

According to the above assumptions, the analytical model can estimate the required network cost for locating a mobile in three strategies. The next subsection explains this model in detail.

### B. The Analytical Model

One common and important term in these three strategies is how long the pointer chain will be. This important term is referred to as  $E[k]$ , i.e., the expected number of forwarding pointers in the locating chain.  $E[k]$  can be derived from the following equation:

$$E[k] = \sum_{K=0}^{\infty} E[k|K] \alpha(K). \quad (1)$$

In the real world, a mobile may move across the RA boundary  $K$  times between two consecutive incoming calls, and the number of resulting forwarding pointers is  $k$ . It should be

understood that  $k$  is not always equal to  $K$ . The exact equation is  $K \geq k$ . The equal sign is true only when a mobile never revisits any RA between two consecutive incoming calls.

$\alpha(K)$  is the probability that a mobile moves across the RA boundary  $K$  times between two consecutive incoming calls. From [8, Appendix],  $\alpha(K)$  can be calculated as follows:

$$\alpha(K) = \begin{cases} 1 - \frac{1 - f_m^*(\lambda_c)}{\theta}, & K = 0 \\ \frac{1}{\theta} [1 - f_m^*(\lambda_c)]^2 [f_m^*(\lambda_c)]^{K-1}, & K > 0 \end{cases} \quad (2)$$

where  $\lambda_c$  is the call arrival rate and  $\lambda_m$  is the frequency of crossing the RA boundary,  $\theta = (\lambda_c/\lambda_m)$  is the call-to-mobility ratio, and  $f_m^*$  is the Laplace-transform of the mobile residential time in an RA [8], [14], [15].

In this model, the residential time is assumed to have a Gamma distribution. This assumption has two explanations. One is that the Gamma distribution can have the desirable property to fit an arbitrary distribution by setting appropriate parameters. Another explanation is that the Gamma distribution has a simple Laplace transform which is as follows:

$$f_m^*(s) = \left( \frac{\lambda_m \gamma}{s + \lambda_m \gamma} \right)^\gamma \quad \text{where } \gamma = \frac{1}{V \lambda_m^2}. \quad (3)$$

By using (2) and (3), different  $\lambda_c$ ,  $\lambda_m$ ,  $V$ , and  $K$  can be set to obtain the corresponding probability  $\alpha(K)$ .

To derive  $E[k]$ , another critical factor is  $E[k|K]$ , which means the expected length of the locating pointer chain for a given  $K$ .  $E[k|K]$  can be computed by a simulation technique called the two-dimensional random walk. In a two-dimensional random walk, a mobile may move in one of four possible directions with equal probability 0.25. However, in the real world, the mobile's moving pattern usually exhibits spatial locality. That is, a mobile tends to roam within a bounded area. To simulate this characteristic, a two-dimensional random walk within a bounded region is used to model a mobile's moving pattern. In this simulation, a mobile's movement is restricted in a limited square area. If a mobile is at the edge of this limited area, the probability for each direction is no longer 0.25. This is owing to the disallowance to move across the limited area's boundary. Fig. 10 illustrates a bounded area with size 16. When a mobile is in RA<sub>1</sub>, it is not allowed to move to the right. If it moves horizontally, it must move to the left. Therefore, the probabilities moving to the left is 0.5 in RA<sub>1</sub>. Based on the same reason, the probabilities moving to the right and up are all 0.5 in RA<sub>2</sub>.

For a given  $K$ , we can determine  $K$  consecutive movement directions with the probability introduced above, and compute the resulting length of the locating pointer chain  $k$ . An example is shown in Fig. 11;  $K = 16$  and  $k = 6$ . This experiment is repeated with the described probability until the  $E[k|K]$  values are convergent. By such simulation,  $E[k|K]$  can be computed for any given  $K$ . Now, the expected length of the locating pointer chain, i.e.,  $E[k]$ , can be computed from (1).

Because the value of  $\alpha(K)$  is significantly affected by  $\theta$ ,  $E[k]$  is a function of  $\theta$ . To express this fact,  $E[k]$  is re-

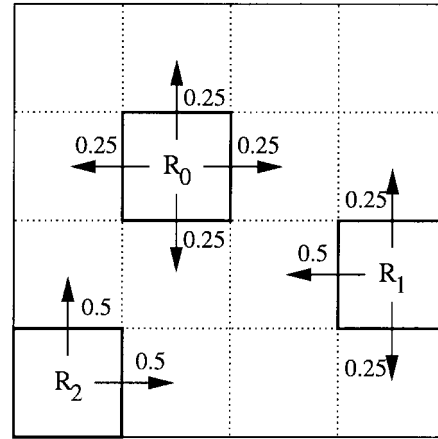


Fig. 10. Two-dimensional random walk within bounded area.

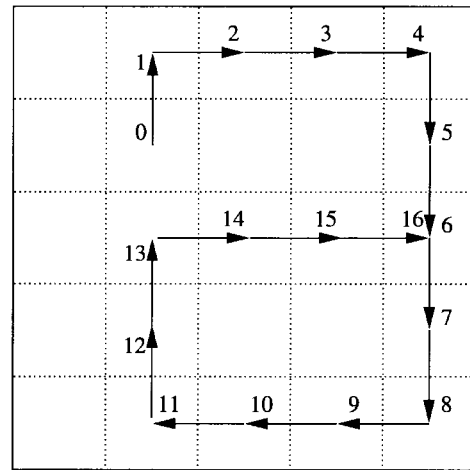


Fig. 11. Example for  $K = 16$  and  $k = 6$ .

expressed as  $E[k|\theta]$ . According to [8], the locating cost in PFSHLR for a call delivery is

$$C_S = 3 \left( \frac{N-1}{N} \right) + 2 \left( \frac{1}{N} \right) + E[k|\theta] \delta \quad (4)$$

where  $N$  is the number of distributed HLR's at remote PSTN's. Based on the assumption that incoming calls are uniformly distributed among  $N$  remote PSTNS's, the locating cost for a call delivery in PFDHLR is

$$C_D = 2 + E \left[ k \left| \frac{\theta}{N} \right. \right] \delta. \quad (5)$$

As mentioned in Section III, OSPFDHLR guarantees that the length of forwarding pointer chain does not exceed one. When an incoming call arrives, the locating path is traced as in Fig. 12. Therefore, for OSPFDHLR, the cost of locating a mobile for a call delivery is

$$C_O = \begin{cases} 2 + \delta, & E \left[ k \left| \frac{\theta}{N} \right. \right] \geq 1 \\ 2 + E \left[ k \left| \frac{\theta}{N} \right. \right] \delta, & E \left[ k \left| \frac{\theta}{N} \right. \right] < 1. \end{cases} \quad (6)$$

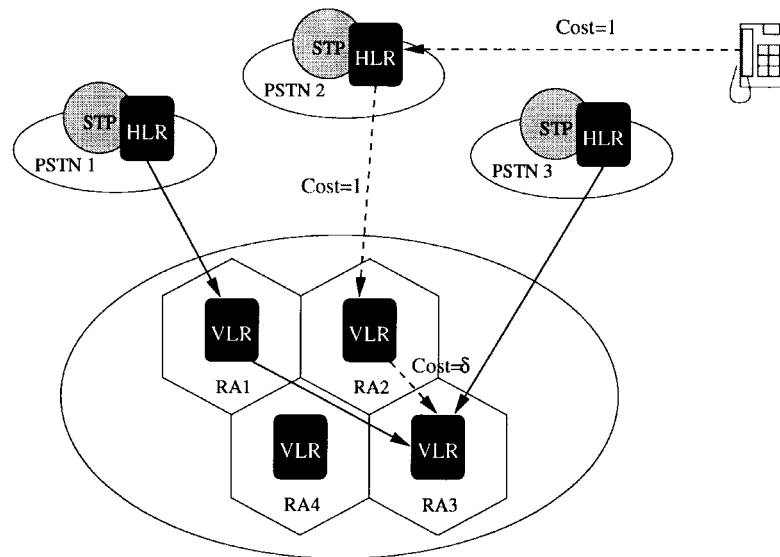


Fig. 12. Process of locating a mobile for call delivery from remote PSTN in OSPFDHLR.

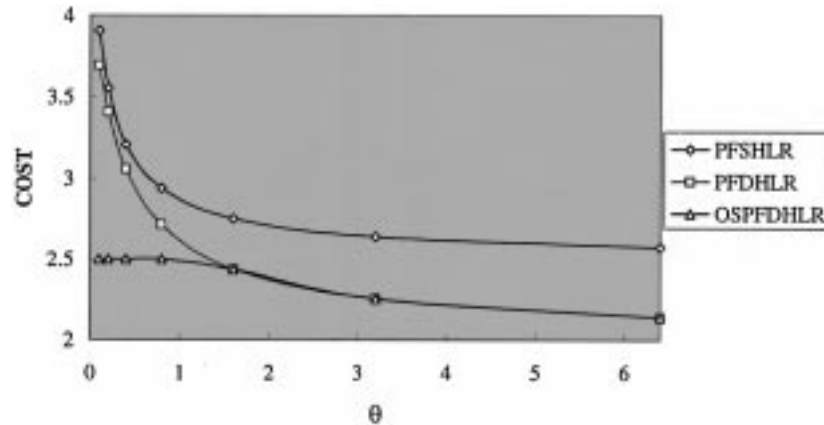


Fig. 13. Locating costs of three strategies with  $S = 16$ ,  $N = 2$ ,  $V = 1.0$ , and  $\delta = 0.5$ .

By using (4)–(6), simulations are performed for these three strategies.

### C. Simulation Results

Five factors influence the simulation results. They are described as follows.

- $N$  Amount of distributed HLR's in the system.
- $S$  Size of the bounded square region in which the mobile is restricted due to spatial locality.
- $V$  Variance of the resident time distribution in this simulation.
- $\delta$  Cost for traversing a forwarding pointer from one VLR to another.
- $\theta$  Call-to-mobility ratio  $\theta = (\lambda_c/\lambda_m)$  where  $\lambda_c$  is the call arriving rate and  $\lambda_m$  is the frequency of crossing an RA boundary.

Figs. 13–18 present the costs for PFSHLR, PFDHLR, and OSPFDHLR with the different parameters mentioned above. To more thoroughly examine how the spatial locality affects the performance of the three strategies, Figs. 19 and 20 provide further exploration.

### D. Numerical Analysis

1) *General Analysis:* Fig. 13 summarizes the simulation results for the per-pointer traversing cost with  $\delta = 0.5$ . The other parameters are the size of the locality restricted area  $S = 16$ , the amount of distributed HLR's  $N = 2$ , and the variance of the resident time distribution  $V = 1.0$ . As shown in Fig. 13, PFDHLR outperforms PFSHLR in such a condition. Another tendency worth noting is that the costs for PFDHLR and PFSHLR rise quickly when  $\theta$  decreases from 0.4. Fig. 14 considers a high per-pointer traversing cost, e.g.,  $\delta = 1.0$ . On average, PFDHLR still outperforms PFSHLR. However, the curve for PFDHLR becomes slightly higher than in PFSHLR for  $\theta \leq 0.8$ . As already mentioned, PFDHLR may have longer forwarding pointer chains due to its smaller call-to-mobility ratio for each distributed HLR.

OSPFDDLRL can overcome this potential problem to make the concept of distributed HLR's work better. Fig. 13 indicates that a 14.56% improvement is made from OSPFDHLR over PFDHLR for  $\delta = 0.5$ . For  $\delta = 1.0$ , a 22.59% improvement can be expected, as Fig. 14 indicates.



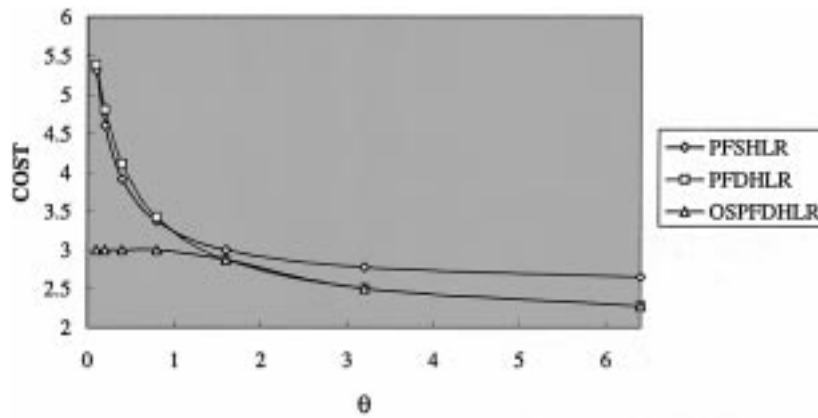


Fig. 14. Locating costs of three strategies with  $S = 16$ ,  $N = 2$ ,  $V = 1.0$ , and  $\delta = 1.0$ .

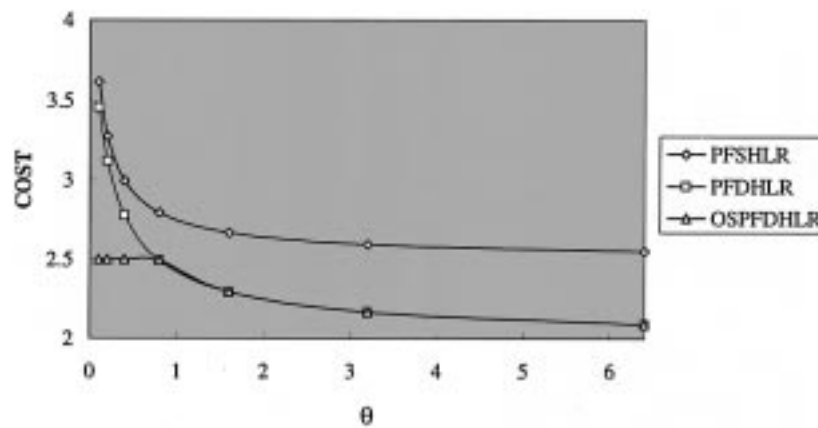


Fig. 15. Locating costs of three strategies with  $S = 16$ ,  $N = 2$ ,  $V = 10.0$ , and  $\delta = 0.5$ .

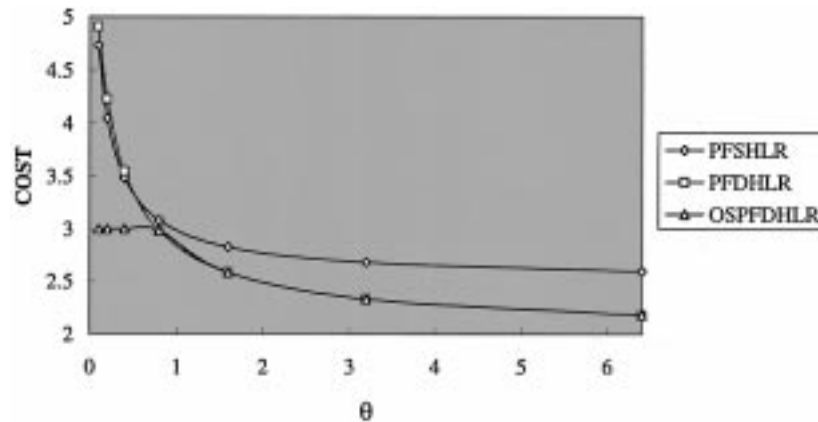


Fig. 16. Locating costs of three strategies with  $S = 16$ ,  $N = 2$ ,  $V = 10.0$ , and  $\delta = 1.0$ .

Figs. 15 and 16 consider the case of residential time with high variance. In these two figures,  $V = 10.0$ , and the other parameters are the same as in Figs. 13 and 14. Comparing Figs. 13 and 15 reveals that the curves are rather similar. The costs in Fig. 15 are a little smaller than that in Fig. 13. The reason for this phenomenon is that  $E[k]$  will be shortened slightly when the variance is increased. This property has been explained in [8]. Therefore, for the residential time with high variance, the costs for PFSHLR and PFDHLR only

slightly decrease. In Fig. 16, the improvements made from OSPFDHLR over PFDHLR are 10.03 and 16.20%.

Figs. 17 and 18 consider more distributed HLR's, e.g.,  $N = 4$ , in the system. Comparing Figs. 13 and 17, PFDHLR still outperforms PFSHLR, but is very close to PFSHLR as  $\theta \leq 0.4$ . The reason is that the call arrival rate is divided by four now due to the assumption that the incoming calls are uniformly shared by distributed HLR's. This division makes  $\theta$  decrease and the expected number of forwarding

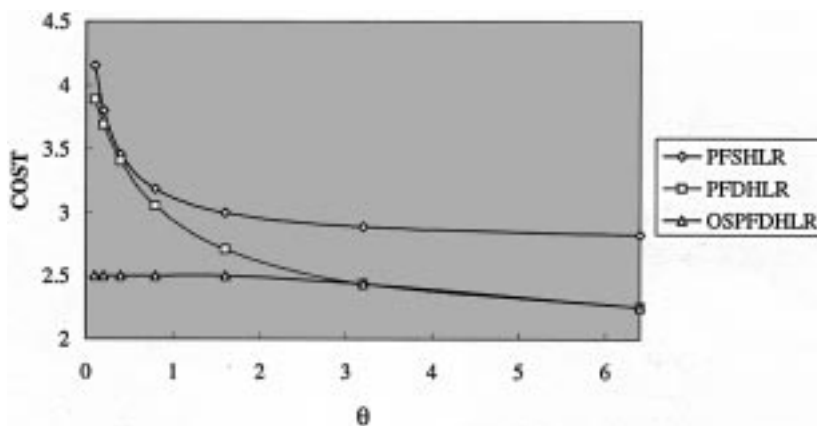


Fig. 17. Locating costs of three strategies with  $S = 16$ ,  $N = 4$ ,  $V = 1.0$ , and  $\delta = 0.5$ .

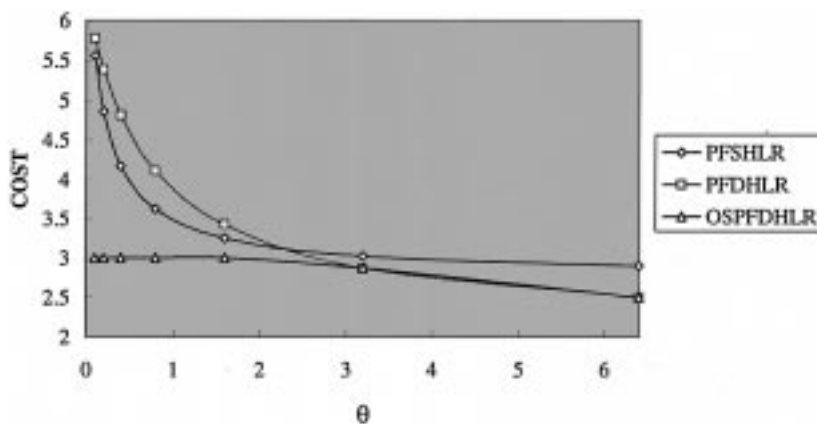


Fig. 18. Locating costs of three strategies with  $S = 16$ ,  $N = 4$ ,  $V = 1.0$ , and  $\delta = 1.0$ .

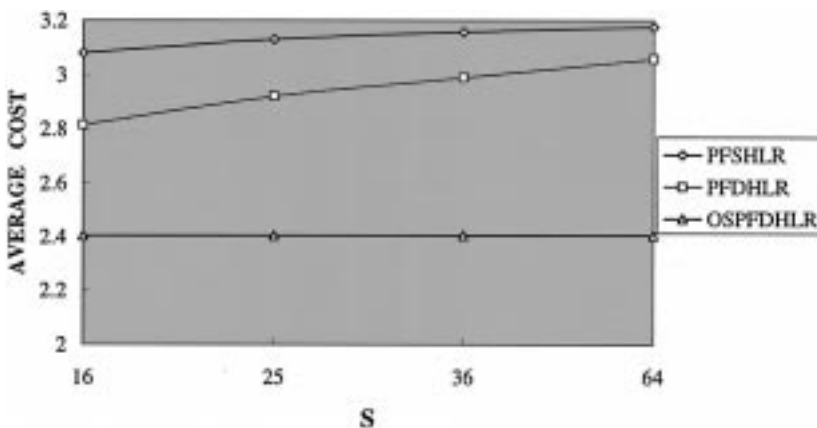


Fig. 19. Average costs for different spatial locality ( $N = 2$ ,  $V = 1.0$ ,  $\delta = 0.5$ ).

pointers increase significantly. When a high traversing cost is considered in Fig. 18, PFDHLR may require a higher cost than PFSHLR if  $\theta < 2.2$ . Under such a condition, the improvement made by OSPFDHLR becomes more significant. They are 19.84 and 29.46% corresponding to Figs. 17 and 18, respectively.

2) *Further Exploration into Spatial Locality:* Figs. 19 and 20 demonstrate how the spatial locality has an influence on

these three strategies. Fig. 19 plots the average costs of the three strategies for different  $S$  values. The values of the other parameters are  $0.1 \leq \theta \leq 6.4$ ,  $N = 4$ ,  $V = 1.0$ , and  $\delta = 0.5$ , as shown in this figure. Fig. 20 considers four distributed HLR's (i.e.,  $N = 4$ ) in the system. Fig. 19 indicates that PFDHLR always outperforms PFSHLR as  $\delta = 0.5$ . The average costs for PFSHLR and PFDHLR all rise slightly, and the curve tends to become a horizontal line with increasing  $S$ .

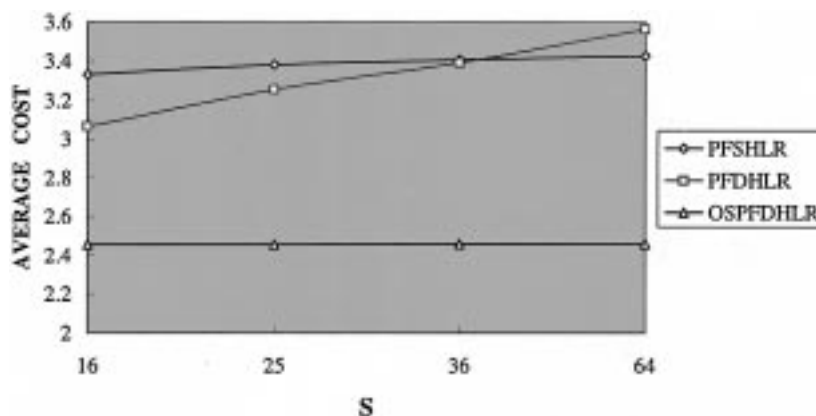


Fig. 20. Average costs for different spatial locality ( $N = 4$ ,  $V = 1.0$ ,  $\delta = 0.5$ ).

Furthermore, if the number of distributed HLR's,  $N$ , increases from 2 to 4, PFSHLR may outperform PFDHLR for  $\delta = 0.5$  and  $S > 36$ , as shown in Fig. 20.

This comparison between PFDHLR and PFSHLR provides a manner of determining which locating strategy is appropriate for subscribers with different spatial localities. For example, PFDHLR should be preferred more than PFSHLR under the condition in Fig. 20 when  $S < 36$ . However, a more ideal choice is OSPFDHLR. As those figures demonstrate, OSPFDHLR always outperforms the other two strategies in terms of performance analysis.

## V. CONCLUSION

Location tracking is an important topic in a PCS system. In IS-41 and GSM, a two-level hierarchical HLR/VLR strategy is used to achieve location tracking. However, HLR is gradually becoming the bottleneck of the signaling network in the system. To distribute HLR's in several regions is a natural solution. But in IS-41 and GSM, this solution needs multiple HLR updates costs when a mobile registers at a new RA. PFDHLR is a good idea to implement distributed HLR's with pointer forwarding to disperse the traffic load at HLR. However, it may be necessary to traverse a long pointer chain to locate a mobile in PFDHLR. This potential problem must be overcome to make the idea of distributed HLR's more feasible and attractive.

In this paper, an efficient location tracking strategy, namely, one-step pointer forwarding with distributed HLR's, is proposed. It can properly adjust the forwarding pointers in VLR's, and guarantees that the length of any pointer chain does not exceed one, regardless of from which distributed HLR a call originates. It only properly adjusts the pointers between VLR's instead of using an expensive HLR access. Simulation results demonstrate that, compared with PFDHLR, OSPFDHLR can reduce the call delivery time/cost significantly, particularly when the number of distributed HLR's is growing.

In fact, OSPFDHLR provides an upper bound of the location tracking time. The reason is that the locating pointer chain does not exceed one in any case. This merit is useful in providing high-quality service in a distributed HLR system. Another salient feature of OSPFDHLR is the ability to reclaim the

invalid entry in VLR's. We believe that the major contribution of this paper is to strengthen the feasibility and attractiveness of the concept of distributed HLR's.

One potential problem in OSPFDHLR is that adjusting the forwarding pointers needs some extra signaling capacity, between VLR's. For a mobile which often moves across the RA boundary but does not frequently receive calls, a considerable signaling capacity will be wasted. For this kind of service subscriber, we suggest a flexible countermeasure, the " $K$ -step pointer forwarding with distributed HLR's" (KSPFDHLR) strategy. In KSPFDHLR, the pointer chain will be adjusted only when its length is more than  $K$ . In fact, OSPFDHLR is a special case of KSPFDHLR,  $K = 1$ . We think that how to find a proper  $K$  to suit subscribers with a certain mobility pattern is a worthwhile research topic in the future.

## ACKNOWLEDGMENT

The authors would like to thank Prof. Y.-B. Lin for the assistance during the course of this research.

## REFERENCES

- [1] G. H. Forman and J. Zahorjan, "The challenges of mobile computing," *Computer*, pp. 38–47, Apr. 1994.
- [2] S. Mohan and R. Jain, "Two user location strategies for personal communications services," *IEEE Personal Commun.*, pp. 42–50, First Quarter 1994.
- [3] J. Z. Wang, "A fully distributed location registration strategy for universal personal communication system," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 850–860, Aug. 1993.
- [4] Y.-B. Lin, "Determining the user locations for personal communications services networks," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 466–473, Aug. 1994.
- [5] R. Jain, Y.-B. Lin, C. N. Lo, and S. Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1434–1445, Oct. 1994.
- [6] Y.-B. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Trans. Networking*, submitted for publication.
- [7] R. Jain and Y.-B. Lin, "An auxiliary user location strategy employing forwarding pointers to reduce network impacts of PCS," *ACM-JCM Wireless Networks*, pp. 197–210, 1995.
- [8] Y.-B. Lin, "Location tracking with distributed HLR's and pointer forwarding," in *Proc. 1995 Int. Symp. Commun.*, 1995, pp. 31–37.
- [9] Y.-B. Lin and S.-Y. Hwang, "Deregistration strategies of PCS networks," in *Proc. ICS'94*, Dec. 1994, pp. 145–149.
- [10] B. C. Kim, J. S. Choi, and C. K. Un, "A new distributed location management algorithm for broadband personal communication networks," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 516–524, Aug. 1995.

- [11] Y.-B. Lin and A. Noerpel, "Implicit deregistration in a PCS network," *IEEE Trans. Veh. Technol.*, vol. 43, no. 4, pp. 1006–1010, 1994.
- [12] B. Jabbari, G. Colombo, A. Nakajima, and J. Kulkarni, "Network issues for wireless communications," *IEEE Commun. Mag.*, pp. 88–98, Jan. 1995.
- [13] Y.-B. Lin and S.-Y. Hwang, "Comparing the PCS location tracking strategies," *IEEE Trans. Veh. Technol.*, submitted for publication.
- [14] W. Feller, *An Introduction to Probability Theory and Its Applications, vol. I*. New York: Wiley, 1966.
- [15] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1983.
- [16] K. L. Sue, "One-step pointer forwarding strategy for location tracking in distributed HLR environment," M.S. thesis, Dept. Comput. Sci. Inform. Eng., Chiao-Tung Univ., Hsinchu, Taiwan, 1996.



**Kuen-Liang Sue** received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1991, and the M.S. degree, also in computer science and information engineering, from National Chiao-Tung University, Hsinchu, Taiwan, in 1996.

He is currently an Instructor in the Department of Information Management, Ling-Tung College, Taichung, Taiwan. His research interests include personal communication services network, mobile computing, and databases.



**Chien-Chao Tseng** received the B.S. degree in industrial engineering from National Tsing-Hua University, Hsinchu, Taiwan, R.O.C., in 1981, and the M.S. and Ph.D. degrees in computer science from Southern Methodist University, Dallas, TX, in 1986 and 1989, respectively.

He is currently a Professor in the Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsinchu, Taiwan. His research interests include mobile computing and parallel and distributed processing.