Wireless Networks 5 (1999) 287–297

A cost effective distributed location management strategy for wireless networks

Kuochen Wang and Jung Huey

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC

The mobility feature of mobile stations (MSs) imposes a large burden on network traffic control as a result of location management. Design issues of location management include *MS registration (updating)* and *call set-up (paging)*. Previous approaches introduced several network topologies for updating and paging procedures, but most of them focused on a single problem: either updating optimization or paging optimization. In this paper, we design and integrate two mechanisms, *distributed temporary location caches* (TLCs) and *distributed home location registers* (HLRs), to reduce database access delay and to decrease network signaling traffic in both updating and paging for low power, low tier micro cellular systems. By using TLCs, our approach can improve the performance of updating and paging in comparison with previous approaches. Experimental results based on our analytic model show that our location management procedures have lower HLR access rate, lower registration cost, and lower call set-up cost than other approaches.

1. Introduction

In recent years, the development of wireless communication networks has grown rapidly. Fixed and wireless networks adopt quite different approaches to network control on end users. A mobile station (MS) may access fixed network services via wireless and fixed networks. Due to the mobility of MSs, high-performance connection set-up controls and location management strategies are necessary [10]. In Internet, we transmit datagrams to a fixed end user based on its IP address. But in a wireless mobile network, we cannot adopt IP communication protocols for mobile users directly. In order to switch incoming calls to an MS efficiently, the MS should report its location to the network periodically. We call this reporting procedure as *MS location registration (updating)*. An MS will perform the registration procedure wherever it enters a new service area. When a call arrives, the network broadcasts this call to one or more registration areas (RAs) to locate a specific MS, where an RA consists of one or more cells, forming a contiguous geographical region [8]. We call this procedure as *call set-up (alerting paging)*. Any time an MS enters a new service area, it informs the network about its new address. This will cause high traffic cost in its registration procedure. However, when a new incoming call arrives, no extra cost is needed because the network already knows exactly the location of each MS anytime. If the whole network belongs to one large service area, an MS can roam around it without any registration operations. But it may result in high overhead because the paging signal has to be broadcast to the whole network. More paging operations mean more network signaling traffic. More updating procedures result in more database access load. Thus, we should select a good MS tracking policy to minimize the total location management cost (MS registration plus call set-up).

As the number of MSs keeps increasing, the cost of updating and paging due to the complexity of network signaling and the delay of database access become undesirable. Our design goal is to reduce the overhead in location tracking procedures. To solve these problems, we integrate two mechanisms. One is *distributed temporary location caches* (TLCs). A TLC has a location cache like a cache in a file access system. It reuses MS location information to simplify the complexity of network signaling. The other is *distributed home location registers* (HLRs). To reduce database access delay, we use only HLRs, which are distributed in the network, for MS registration and call set-up without any traditional visitor location register (VLR).

The remainder of this paper is organized as follows. Section 2 gives an overview of existing approaches. We propose our location techniques including network architecture, MS registration (updating), call set-up (paging), and the associated TLC activities in section 3. In section 4, we give a formal representation of location management protocols. Performance analysis and experimental results are given in section 5. Finally, we make a few concluding remarks in section 6.

2. Related work on location management

Due to large demand for wireless communications and the MS mobility feature, location management becomes an important issue. As mentioned before, location management techniques focus on two major tasks: *MS registration* and *call set-up*. MS registration involves updating location databases when current location information is available, while call set-up involves querying of location databases to determine the current location of the called MS [1]. Existing location management techniques can be classified as follows: centralized database architectures and distributed database architectures [1].

Centralized database architecture keeps the original network topology unchanged. They use the location procedures modified from IS-41 or GSM, or add some functions to the IS-41 or GSM network devices to improve network performance. Distributed database architecture completely changes the network topology. It distributes subscribers profile to several databases and provides a new algorithm for location management. We will review some representative algorithms for each case.

2.1. Centralized database architecture

Ref. [12] provides a simple approach to locate MSs. It uses a central database (HLR) to maintain the location information of all MSs in the network. Although its algorithm and implementation are simple, when the number of mobile users keeps increasing, it would be hard to store all MS's information in the central database. It will also increase database load if multiple users access the only database simultaneously, and the network would be down when the database crashes.

Current researches on location techniques are almost based on the standard, IS-41 [6] or GSM [13,15]. Both standards are based on two-level databases, HLR and VLRs, to maintain the MS's location information. A *per-user location cache* scheme was studied in [8]. It maintains a local storage or a cache of user profile at the STP (signal transfer point [14]). A cache entry is added whenever an MS is called through the STP. When another call arrives for that MS, the STP first checks the cache to find the called MS's location information. If it is a cache hit, the STP directly transmits the call to the destination STP and completes the call. Because the called MS may move to another service area which is not associated with the cache, the IS-41 call set-up scheme is used to locate the MS in this situation. In this case, it not only cannot improve the performance of call set-up, but also can increase its complexity. If there is no entry in the cache for the called MS, it would perform an IS-41-like call set-up procedure to locate the MS. Although the per-user cache can reduce network signaling traffic and the number of database accesses during the call set-up, when an MS registers, it still needs to access HLR and VLRs.

2.2. Distributed database architecture

Ref. [10] removes all VLRs and only retains HLRs for location information storage, which manage MSs initially registered in each access network. Whenever an MS moves into a different RA, a registration message is delivered to its HLR to update the current location. At call set-up, location information is retrieved once from the MS's HLR to determine the current location of the MS [10]. This scheme reduces the number of database accesses for mobility management at the cost of some increase of network signaling traffic [10]. It assumes that there will be enough capacity of signaling network with the advances in fiber optic technology [10]. However, with the prosperity of multimedia applications, the capacity of signaling network is always a concern. Our approach not only reduces the number of database accesses, but also decreases network signaling traffic. A *hierarchical location registration* scheme is introduced in [16]. Its database architecture is a tree. MS locations are kept by the database at the leaves. An MS entry is created in each database from a leaf to the root. The root (highest layer) may be the earth followed by country, state, etc. The leaf (layer one) may be an RA [16]. Whenever a call is initiated, it just reaches any one database in that path, and knows the called MS's address. This scheme can reduce signaling message traveling distance, but it increases multiple database access and query operations when it performs MS registration or call set-up. A *location database partition* scheme is proposed in [2]. It distributes all its databases in the network, and the architecture is also similar to a tree. It collects mobility patterns and partitions the location database into several groups. If an MS moves around in the same group, it does not need to register. This algorithm minimizes the number of registration operations with a high mobility rate. However, before constructing the databases, MS moving and call arriving patterns are required in advance, and it still has the same disadvantage as [16] when it performs call set-up.

3. Design approach

In this section, we propose new location management protocols. We first show the proposed network architecture. We utilize the user locality property to simplify location techniques. User locality in a mobile computing environment can be looked at from both spatial and temporal points of view, and with respect to the movements of MSs and the calls that are made to them [4].

3.1. Network architecture

In our network topology, there are two layers of networks: *access networks* and *transport networks* [10], as shown in figure 1. The transport network is based on the asynchronous transfer mode (ATM) technology to provide one access network to connect to another access network. The access network is the interface between MSs and the wireless network [13]. Different access networks are connected via a transport network. Each access network includes some components, like a *home location register* (HLR), a *temporary location cache* (TLC), *mobile switching centers* (MSCs), *base station controllers* (BSCs), *base stations* (BSs), and *mobile stations* (MSs). Our approach does not require any VLR. The functions provided by the VLR and MSC are now supported by the TLC and MSC. A TLC is a dynamic cache, whose entries change with mobile users. It maintains a local storage of user location information and issues the MS registration and call set-up messages. Note that our approach is more suitable for a low power,

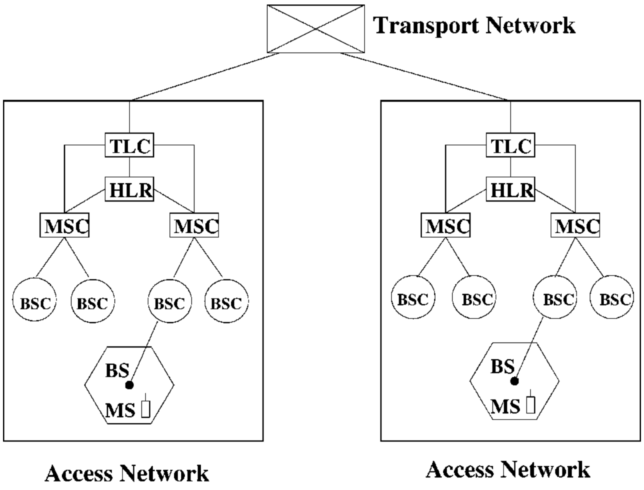

Figure 1. Overall network topology.

low tier micro cellular system, where a TLC is expected to have a high hit ratio due to user locality during MS registration and call set-up. MSCs provide typical switching functions and coordinate location registration and call set-up procedures. An MSC handles MS registration, performs call set-up, and forwards TLC queries. That is, the MSC's functions are modified such that it talks to the TLC directly.

3.2. MS registration (updating)

In order to deliver calls, the network must keep the location information of each MS. The data in the databases become incorrect as an MS moves around the network. So, the location information stored in the databases should be updated when an MS moves out of its current service area. When an MS moves, traditional registration procedures would need to coordinate between HLR and VLRs. In addition, it generates high signaling traffic due to database accesses. We improve the registration procedure to avoid these disadvantages. In our algorithm, we divide the registration procedure into two types: *intra-access network* (intra-AN) *registration* and *inter-AN registration*.

3.2.1. Intra-AN registration

As an MS moves around an access network, it would pass through many RAs. Whenever an MS moves to a new RA in the same access network, it only informs its location to the local TLC and updates its associated entry about its RA identifier (RAI). We use a TLC not only for registration, but also for routing purpose. An MS registers at its associated MSC by sending the *upd_req*(TMSI,RAI,*acc_no*) message containing the following identification data: temporary mobile subscriber identity (TMSI), new RA identifier (RAI), access network number (*acc_no*), and some authentication data. A TMSI consists of an HLR's address and a ciphered MS identifier, where the former is for finding the MS's home HLR and the latter is used for finding the MS entry in its home HLR [10]. Figure 2 depicts the relationship between each network device. The intra-AN registration procedure in figure 3 is described as follows:

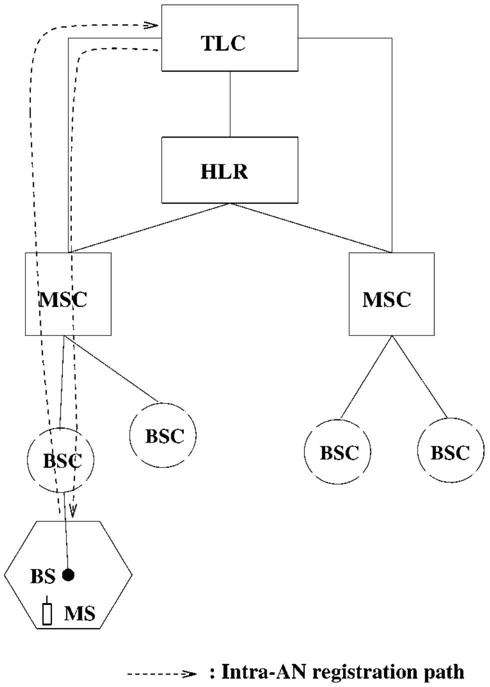

Figure 2. Intra-AN registration in an access network.

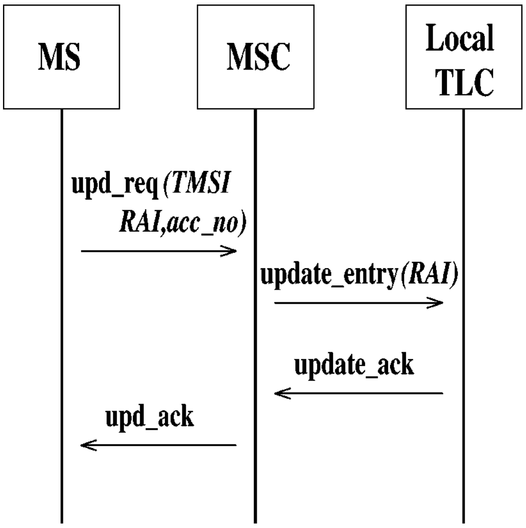

Figure 3. Signal flow of intra-AN registration.

1. When an MS enters a new RA in the same access network, it issues a registration request, *upd_req*(TMSI,RAI, *acc_no*), to the associated MSC.
2. The MSC forwards this request to the local TLC and the local TLC determines that it is an intra-AN registration by *acc_no* and then executes *update_entry*(RAI).
3. The corresponding TLC entry of that MS is updated and *update_ack* is sent back to the MSC.
4. The MSC informs the MS of successful intra-AN registration by sending *upd_ack*.

If an MS stays in an access network, the cost due to the registration procedure can be reduced because the MS does

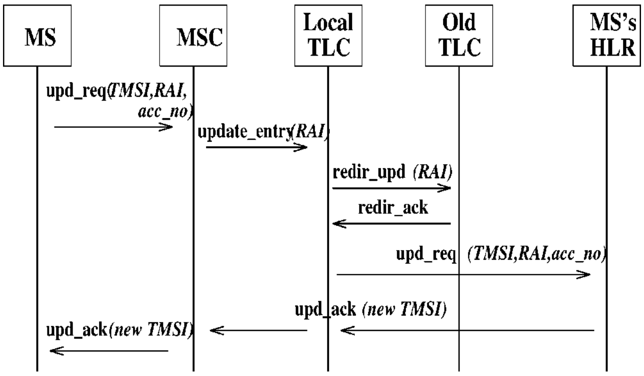

Figure 4. Signal flow of inter-AN registration.

not need to register at its home HLR. This is especially useful when an MS moves far away from its home access network and stays in a remote access network.

3.2.2. Inter-AN registration

When an MS roams around the whole network, it may pass through many access networks. The signal flow of our inter-AN registration procedure in figure 4 is described as follows:

1. When an MS enters a new RA, not in the same access network, it issues a registration request, *upd_req*(TMSI, RAI,*acc_no*) to the associated MSC.
2. The MSC forwards this request to the local TLC. The local TLC determines that it is an inter-AN registration by *acc_no* and then executes *update_entry*(RAI). It updates an entry for that specific user and forwards *upd_req*(TMSI,RAI,*acc_no*) to the MS's home HLR. The local TLC also asks the old TLC that the MS last visited to insert a redirection pointer by issuing *redir_upd*(RAI).
3. The HLR entry of that MS is updated and, if successful authentication, it then returns an acknowledgement, *upd_ack*(*new TMSI*), to the MSC through the local TLC.
4. The MSC informs the MS of successful inter-AN registration by forwarding *upd_ack*(*new TMSI*).

The redirection pointer in the old TLC helps to decrease the miss rate during call set-up. In the MS registration procedures (intra-AN and inter-AN registration) mentioned above, our scheme has less database access delay than previous strategies.

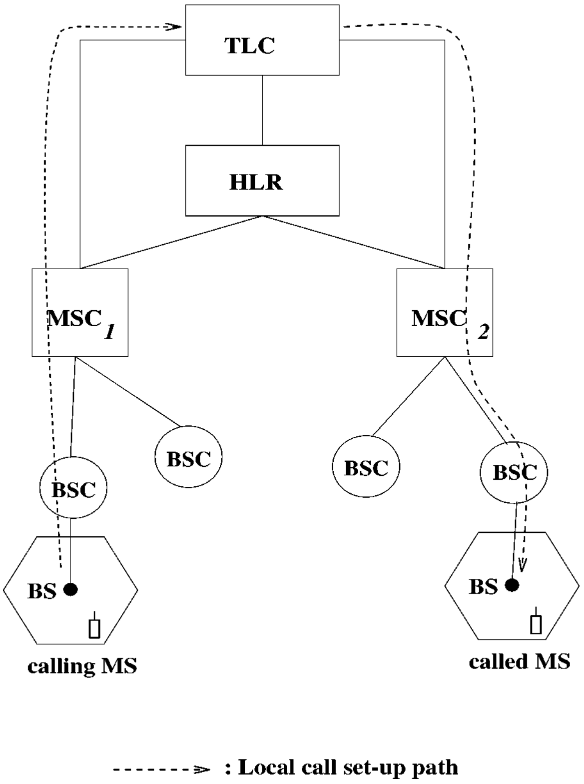

Figure 5. Local call set-up in an access network.

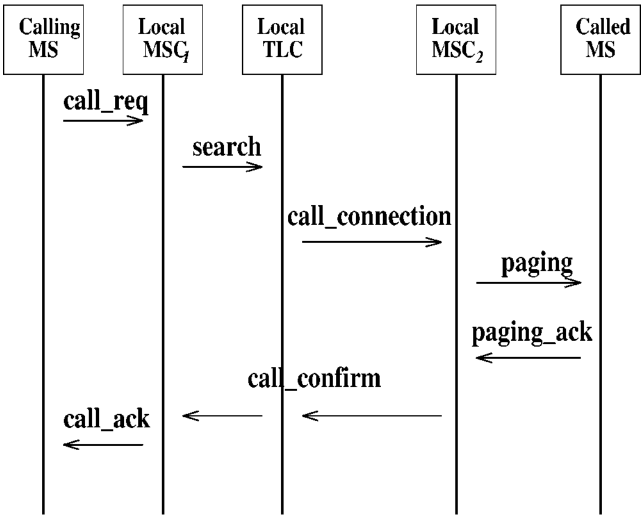

Figure 6. Local call set-up.

3.3. Call set-up (paging)

In our proposed call set-up, we divide our call set-up procedure into two types: *local call set-up* and *remote call set-up*, and illustrate them as follows.

3.3.1. Local call set-up

If the calling party and the called party are in the same access network, it is a *local call set-up*. Figure 5 shows the call set-up path between two MSs. In figure 6, we only need to search the address of the called MS in the local TLC. The detailed steps are described as follows:

1. When an MS initiates *call_req* to the associated MSC, the MSC would issue a *search* request to the local TLC.
2. The local TLC has the called MS's address information since the calling and called MSs are in the same access network. The local TLC forwards *call_connection* to the called MSC.

3. The called MSC pages the called MS and waits for its acknowledgment. If the called MS is not in use, it responds *paging_ack* to the called MSC.
4. When the called MSC receives responses, it issues *call_confirm* to the calling MSC through the local TLC.
5. The calling MSC informs the calling MS that the call request is successful by sending *call_ack*.

The *local call set-up* completes a call fast, and it needs less signaling traffic and fewer database accesses. During the MS registration, the local TLC will update the MS location information. When a call request arrives, it only needs to access the local TLC to get the address of the called MS. This way is more cost effective than accessing the HLR, which may be far away.

3.3.2. Remote call set-up

A remote call set-up situation occurs when the calling and called parties are in different access networks. There are *hit* and *miss* cases in our remote call set-up. Figure 7 shows the *miss* case of a remote call set-up. It means that the TLC does not have the called MS's location information entry and the calling MS should query the called home HLR to retrieve the location information of the called MS. If the called MS is steady and is requested again later, its address can be found in the local TLC. We call this as the *hit* case of a remote call set-up as shown in figure 8. In the *hit* case, database access delay can be reduced and a connection can be established right away. We summarize the *hit* and *miss* cases as follows:

1. When an MS initiates *call_req* to the MSC, the MSC would issue *search* to the local TLC.
2. If no called MS location entry is found in the local TLC (the *miss* case), the local TLC queries the called MS's home HLR to get its routing address. The HLR returns the routing address to the local TLC. The local TLC adds an entry of the called MS. The local TLC then issues *call_connection* to the remote TLC, and the remote TLC passes this message to the remote MSC.
3. If the local TLC has the called MS location information (the *hit* case), it retrieves the routing address from the local TLC without accessing the called MS's home HLR, and the local TLC issues *call_connection* to the remote TLC, and the remote TLC passes this message to the remote MSC.
4. The remote MSC pages the called MS and waits for its acknowledgment. If the called MS is not in use, it responds *paging_ack* to the remote MSC.
5. When the remote MSC receives *paging_ack*, it issues *call_confirm* to the calling MSC through the local TLC.

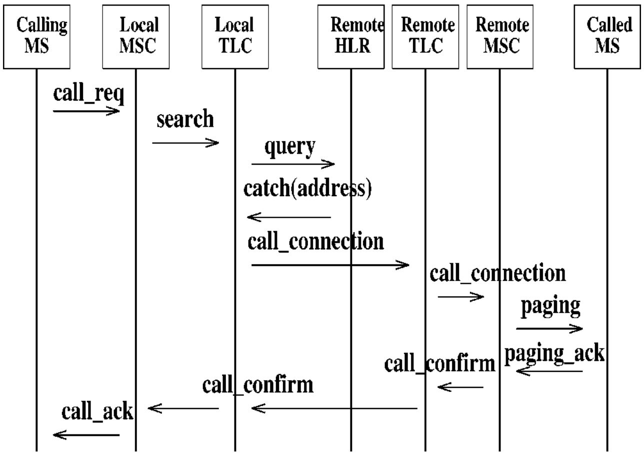

Figure 7. Remote call set-up (miss).

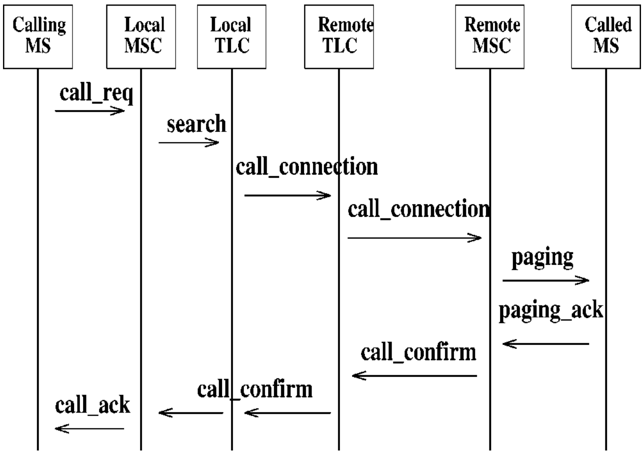

Figure 8. Remote call set-up (hit).

6. The calling MSC informs the calling MS that the call request is successful by issuing *call_ack*.

There is another *miss* situation where an MS moves between two consecutive calls. This will cause a null paging in figure 8. We call this as a *null miss*. To solve this problem, we insert a *redirect pointer* in the old TLC as inter-AN registration and redirect the connection request to the access network where the called MS is visiting. The idea is similar to the *forwarding pointer* introduced in [9]. If a call arrives at a TLC that the MS last visited, the old TLC will redirect the call to the remote TLC according to the redirection entry in it. Figure 9 shows the signaling message flow in the *null miss* case. We see that the *null miss* case is similar to the *hit* case with additional signals between the new and old TLCs. In the null miss case, the local TLC will get a location entry from the old TLC so that subsequent calls can be forwarded directly from the local TLC to the remote TLC. Based on the user locality property, our call set-up procedure can reduce network signaling traffic and database access delay by using TLCs. This is especially useful in the *hit* case of a remote call set-up.

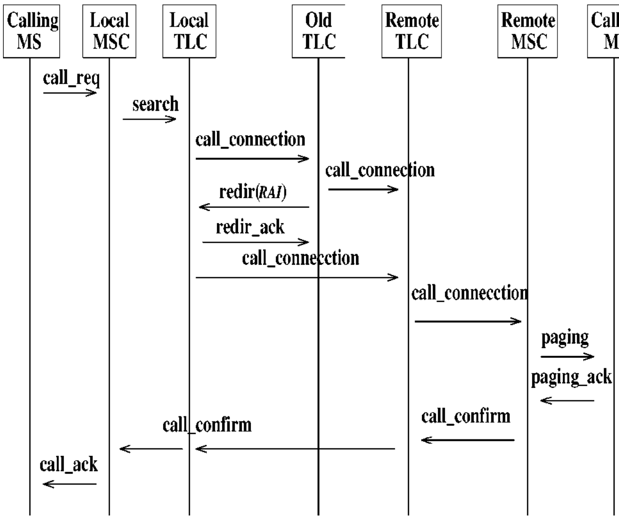

Figure 9. Remote call set-up (null miss).

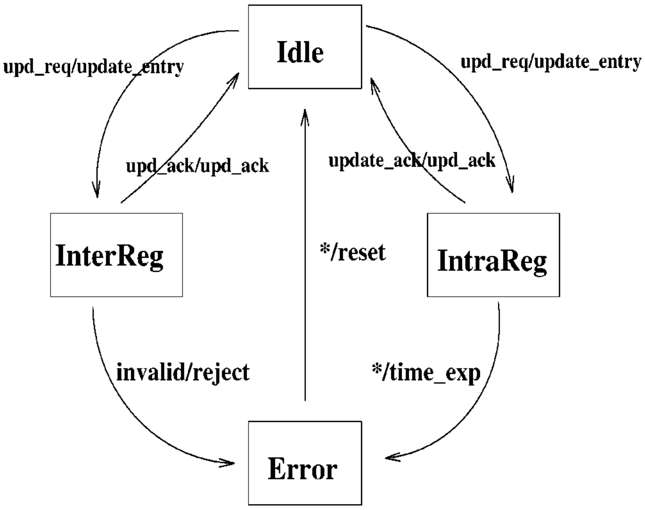

Figure 10. FSD of an MSC during MS registration.

4. Formal representation of location management protocols

When an MS enters a wireless network, it will perform some procedures (ex. MS registration or call set-up) to handle its activities. Since every message will pass through the MSCs, we will define the finite state diagram (FSD) of an MSC to illustrate our location management protocols. In the following figures, we use the notation X/Y to mean that message X is received, and message Y is sent by the MSC [5]. We also define a symbol "$*$" to mean that there is no sending or receiving message in this state. We divide the MSC FSD into two parts, and describe them below.

4.1. FSD of an MSC for signaling flow during MS registration

The FSD of an MSC for signaling flow during MS registration is shown in figure 10. There are four states described as follows:

- **Idle state** It means that the MSC does nothing, and it just maintains the connections to BSCs, the TLC, and the HLR.
- **InterReg state** The MS issues an inter-AN registration request, and the MSC goes to this state.
- **IntraReg state** The MS issues an intra-AN registration request, and the MSC goes to this state.
- **Error state** If the MSC does not receive any response in a limited time (timer expired) or the registration request is invalid, it will be in this state.

4.1.1. MSC state transitions

The state transitions are described as follows:

1. State **Idle**.
   - to **IntraReg**: (*upd_req/update_entry*): The MSC receives *upd_req*(TMSI,RAI,*acc_no*) request and sends *update_entry*(RAI) for updating location information in the local TLC. Based on *acc_no* in *upd_req*, the MSC can distinguish which state (**IntraReg** or **InterReg**) to go.
   - to **InterReg**: (*upd_req/update_entry*): The MSC receives *upd_req*(TMSI,RAI,*acc_no*) request from the MS and goes to this state by *acc_no* in *upd_req*. Also, the MSC issues *update_entry*(RAI) to inform the local TLC to update the MS location entry.
2. State **IntraReg**.
   - to **Idle**: (*update_ack/upd_ack*): If the MSC receives *update_ack*, it means the intra-AN registration is successful. It then issues *upd_ack* to the MS to inform the request is completed.
   - to **Error**: ($*$/*time_exp*): Every time the MSC issues *update_entry*(RAI), it will initialize a timer. The MSC should receive a registration response before the timer expires. If the timer expires and the MSC did not receive any response, the MSC will go to the **Error** state and issues *time_exp* to inform the MS.
3. State **InterReg**.
   - to **Idle**: (*upd_ack/upd_ack*): If an MS performs successful inter-AN registration, the MSC will receive *upd_ack* with a new *TMSI* and forwards it to the MS.
   - to **Error**: (*reject/invalid*): To screen invalid users, the HLR will check if the MS is legal by *upd_req*. If the MSC receives *invalid*, it will go to the **Error** state and then issues *reject* to the MS.
4. State **Error**.
   - to **Idle**: ($*$/*reset*): When the MSC is in the **Error** state, it would issue *reset* to the MS to re-register.

4.2. FSD of an MSC for signaling flow during call set-up

The FSD of an MSC for the signaling flow during call set-up is shown in figure 11. We will focus on the calling

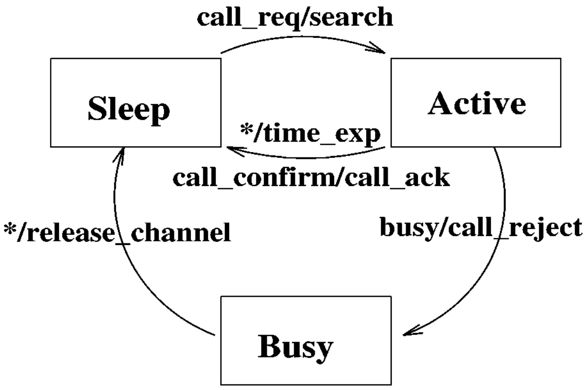

Figure 11. FSD of an MSC during call set-up.

MSC and in the following, “the MSC” means the calling MSC. The FSD for signaling flow during call set-up includes three states:

- **Sleep state** Its activities are the same as the **Idle** State in the MS registration.
- **Active state** The MSC waits for some responses (e.g., *call_confirm, busy, or time_exp*), while the MS issues *call_req*.
- **Busy state** If the called MS is busy or there is no response, the MSC will go to this state.

4.2.1. MSC state transitions

The state transitions are described as follows:

1. State **Sleep**.
   - to **Active**: (*call_req/search*): Whenever the MSC receives *call_req*, it will go to the **Active** state and issue *search* request to the local TLC to search the called MS’s address.
2. State **Active**.
   - to **Busy**: (*busy/call_reject*) *or* (*∗/time_exp*): If the called MS is busy and receives *call_connection*, it will go to the **Busy** state. In this situation, the MSC also sends *call_reject* to the MS.
   - to **Sleep**: (*call_confirm/call_ack*): In the **Active** state, the MSC waits till it gets *call_confirm*. Then, the MSC sends *call_ack* to the MS to indicate that the call request is successful. Every time the MSC issues a *search* request to the local TLC, it should receive responses in a limited time. If the MSC does not receive responses before the timer expires, it will go back to the **Sleep** state and issue *time_exp* to the MS.
3. State **Busy**.
   - to **Sleep**: (*∗/release_channel*): Once the MSC is in this state, it means that the call request is not completed or unsuccessful. The MSC would send *release_channel* to the MS to release the reserved resources and go back to the **Sleep** state to wait for the next request.

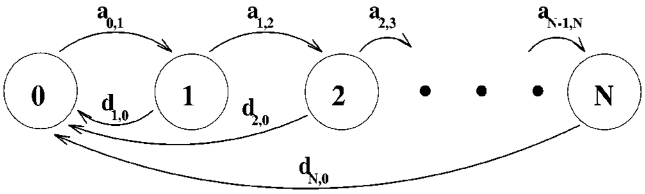

Figure 12. State transition diagram for an imbedded Markov chain.

5. Performance analysis and experimental results

We develop an analytic model to study the performance of our proposed location management protocols. To evaluate, it is necessary to model the user behavior in our network architecture. The user behavior can be classified into two parts: (1) the *user mobility behavior model*, which describes the way a user moves around the network, and (2) the *call set-up model*, which describes how the call is set up when there is an incoming call for a user using our cache scheme (TLC).

Parameters used in our analytic model are summarized as follows:

- C_{L}: cost of transmitting signals via a local link (in an access network).
- C_{R}: cost of transmitting signals via a remote link (in a transport network).
- H_{C}: cost of querying or updating an HLR.
- T_{C}: cost of querying or updating a TLC.
- δ_{TLC} : TLC hit ratio.

5.1. User mobility behavior model

We model the activities of the MS registration using an *imbedded Markov chain* where state i ($i \geqslant 0$) is defined as the number of RAs, i, that the MS has passed by. The state transition diagram for the imbedded Markov chain is given in figure 12. State N is defined as an MS has passed through N RAs in one access network. The state transition $a_{i,i+1}$ ($0 \leqslant i < N$) represents the MS moving rate (from state i to state $i+1$) to the neighboring RA in the same access network. The transition $d_{i,0}$ ($1 \leqslant i \leqslant N$) represents the MS moving rate (from state i to state 0) to another RA out of the access network.

In [3], the border crossing rate out of a given circular area for a single moving user is given by

$$\sigma = 2\frac{v}{\sqrt{\pi a}}, \tag{1}$$

where a is the circular area size. We will evaluate our model using the above equation. Assuming the shape of an RA is circular, we apply equation (1) to an access network containing M equally large RAs and each with size a_{RA}. The border crossing rate γ for an MS out of an RA is

$$\gamma = 2\frac{v}{\sqrt{\pi a_{\mathrm{RA}}}} \tag{2}$$

and the border crossing rate μ for an MS out of an access network is

$$\mu = 2\frac{v}{\sqrt{\pi M a_{\mathrm{RA}}}} = d_{i,0}, \quad \text{for } 1 \leqslant i \leqslant N. \tag{3}$$

Note that an MS that crosses an access network border will also cross an RA border. So, if we want to obtain the rate for RA crossings for which the MS still stays in the access network, the following is obtained from equations (2) and (3):

$$\lambda = \gamma - \mu = \gamma\frac{\sqrt{M}-1}{\sqrt{M}} = a_{i,i+1}, \quad \text{for } 0 \leqslant i < N. \tag{4}$$

The parameters λ and μ are used in our Markov process. We consider P_k which is the equilibrium probability of being in state k of the process. Thus we get

$$\begin{aligned}&(\lambda+\mu)P_k = \lambda P_{k-1}, \quad 0 \leqslant k \leqslant N,\\ &\lambda P_0 = \mu(P_1 + P_2 + \cdots + P_N).\end{aligned} \tag{5}$$

Let us now apply the *z-transform* [11] method. As usual we define

$$P(Z) = \sum_{k=0}^{\infty} P_k Z^k. \tag{6}$$

We then multiply by Z^k, sum, and then identify $P(Z)$ to obtain in the usual way

$$(\lambda+\mu)\sum_{k=1}^{\infty} P_k Z^k = \lambda\sum_{k=1}^{\infty} P_{k-1} Z^k. \tag{7}$$

Solving $P(Z)$ we have

$$P(Z) = \frac{(\lambda+\mu)P_0}{\lambda+\mu-\lambda Z} = \frac{P_0}{1-\frac{\lambda}{\lambda+\mu}Z}. \tag{8}$$

We may now invert equation (8) by inspection [11] to obtain the solution P_k, namely,

$$P_k = P_0\left(\frac{\lambda}{\lambda+\mu}\right)^k, \quad 0 \leqslant k \leqslant N. \tag{9}$$

As we know, the sum of probabilities of all states is "1", so we have

$$\sum_{k=0}^{N} P_k = 1. \tag{10}$$

In order to solve for P_0 we use equations (9) and (10) to obtain

$$P_0 = \left[1 + \sum_{k=1}^{N}\left(\frac{\lambda}{\lambda+\mu}\right)^k\right]^{-1} = \frac{1-\frac{\lambda}{\lambda+\mu}}{1-\left(\frac{\lambda}{\lambda+\mu}\right)^{N+1}}. \tag{11}$$

Thus, finally,

$$P_k = \begin{cases}\dfrac{1-\frac{\lambda}{\lambda+\mu}}{1-\left(\frac{\lambda}{\lambda+\mu}\right)^{N+1}}\left(\dfrac{\lambda}{\lambda+\mu}\right)^k, & 0 \leqslant k \leqslant N,\\ 0, & \text{otherwise.}\end{cases} \tag{12}$$

We use C_{intra} and C_{inter} to represent the cost of intra-AN and inter-AN registration, respectively. Based on equation (12), the average registration cost (C_{reg}) of our approach after an MS has crossed N RAs is

$$\begin{aligned}C_{\text{reg}} &= P_0 C_{\text{inter}} + P_1 C_{\text{intra}} + 2P_2 C_{\text{intra}} + \cdots + NP_N C_{\text{intra}}\\ &= P_0 C_{\text{inter}} + C_{\text{intra}}(P_1 + 2P_2 + \cdots + NP_N)\\ &= P_0 C_{\text{inter}} + C_{\text{intra}}\left(\sum_{k=1}^{N} kP_k\right).\end{aligned} \tag{13}$$

According to our network architecture, C_{intra} and C_{inter} costs are

$$C_{\text{intra}} = 2C_{\mathrm{L}} + T_{\mathrm{c}}, \tag{14}$$

$$C_{\text{inter}} = 4C_{\mathrm{L}} + 2C_{\mathrm{R}} + H_{\mathrm{C}} + T_{\mathrm{C}}. \tag{15}$$

We will compare our mobility model with [10], which has been reviewed in section 2.2 and is referred to as DHLR. In the DHLR approach, a registration message is transmitted to the MS's MSC and then is forwarded to the MS's HLR to update its current location [10]. Next, a registration response message is sent back to the MS. So the average registration cost of DHLR is

$$C_{\text{reg}} = 2C_{\mathrm{L}} + 2C_{\mathrm{R}} + H_{\mathrm{C}}. \tag{16}$$

5.2. Call set-up model

In this section, we will investigate the effect of TLCs which yields net reduction in network signaling and database loads. We use CMR (*call-to-mobility ratio*) [8], which is the average number of calls to an MS from an AN per unit time divided by the average number of times that an MS changes RAs per unit time. For each user, the TLC hit ratio will affect the amount of savings. The entry of a called party is associated with its registration and call set-up behavior, which tends to have a high TLC hit ratio due to user locality.

First of all, based on our remote call set-up steps, we divide them into hit and miss cases. The cost in the hit case is C_{h}, and the cost in the miss case is C_{m}. We get the total cost of call set-up (C_{call}) as follows:

$$C_{\text{call}} = \delta_{\text{TLC}} \times C_{\mathrm{h}} + (1-\delta_{\text{TLC}}) \times C_{\mathrm{m}}. \tag{17}$$

The cost of C_{h} is obtained based on the proposed architecture and protocol. Therefore,

$$C_{\mathrm{h}} = C_{\mathrm{L}} + T_{\mathrm{C}}. \tag{18}$$

Since the miss case involves querying the TLC and the HLR, we have

$$\begin{aligned}C_{\mathrm{m}} &= (C_{\mathrm{L}} + T_{\mathrm{C}}) + (2C_{\mathrm{L}} + 2C_{\mathrm{R}} + H_{\mathrm{C}} + T_{\mathrm{C}})\\ &= 3C_{\mathrm{L}} + 2T_{\mathrm{C}} + 2C_{\mathrm{R}} + H_{\mathrm{C}}.\end{aligned} \tag{19}$$

From equations (17)–(19), the average call set-up cost of our approach is

$$\begin{aligned} C_{\text{call}} &= \delta_{\text{TLC}}(C_{\text{L}} + T_{\text{C}}) \\ &\quad + (1 - \delta_{\text{TLC}})(3C_{\text{L}} + 2T_{\text{C}} + 2C_{\text{R}} + H_{\text{C}}) \\ &= (3C_{\text{L}} + 2T_{\text{C}} + 2C_{\text{R}} + H_{\text{C}}) \\ &\quad - \delta_{\text{TLC}}(2C_{\text{L}} + 2C_{\text{R}} + T_{\text{C}} + H_{\text{C}}). \end{aligned} \tag{20}$$

In the DHLR approach, a call set-up message is transmitted to the called MS's HLR for getting the called MS's current location. Then, the called MS's location is transmitted to the calling MSC [10]. So, the average call set-up cost of DHLR (C_{call}) is

$$C_{\text{call}} = 2C_{\text{L}} + 2C_{\text{R}} + H_{\text{C}}. \tag{21}$$

Now we consider the *CMR* factor. In the equilibrium state, the *CMR* is

$$CMR = \frac{\alpha}{\beta},$$

where α is the mean call arrival rate, and β is the time the MS resides in an RA that has mean $1/\beta$. In [8], the hit ratio of a cache

$$\delta = \frac{\alpha}{\alpha + \beta}, \tag{22}$$

and we can derive *CMR*, assuming incoming calls are a Poisson process and inter-move times are exponentially distributed. Thus,

$$CMR = \frac{\delta}{1 - \delta}. \tag{23}$$

5.3. Numerical results

The following statements for low power, low tier micro cellular systems are assumed to give a reasonable representation of the average behavior of MSs:

1. The average speed of an MS is 5.6 km/h [7].
2. An RA includes 14 cells.
3. The shape of a cell is circular, and its radius is 200 m.
4. A cell can support at most 50 populations.
5. The ratio between T_{C} and H_{C} is $1/5$.

We will evaluate our user mobility model based on the above assumptions. The impact on the load and delay of an HLR is defined by the rate of HLR updates. Figure 13 shows the relation between the mean HLR update rate (U_{h}) and the AN crossing rate (P). Note that P is $\mu/(\lambda + \mu)$. To decrease HLR loads and signaling transmitting distance, we should keep the HLR update rate low. In figure 13, we see that as the AN crossing rate varies from 0.15 to 1, the mean HLR update rate of our proposed protocols is always smaller than that of DHLR except when $P = 1$. In this case ($P = 1$), it means that there is only one RA in the access network. That is, RA border crossings also cause access network border crossings, so that the HLR update rate of our proposed registration protocol is equal to that of DHLR. The smaller HLR update rate of our registration protocol results in less HLR delay and network signaling traffic in comparison with that of DHLR. In DHLR, when an MS moves out of one RA, it always updates its home HLR. In our registration protocol, only crossing access network borders needs to update the HLR. The comparison between our registration cost and that of DHLR is shown in figures 14 and 15. We consider the cases of $C_{\text{intra}}/C_{\text{inter}} = 1/2$, $1/5$, and $1/10$. Figure 14 shows the average registration cost between our approach and the DHLR under various numbers of RA crossings (N). Our approach is always better than DHLR under different $C_{\text{intra}}/C_{\text{inter}}$. In the following figures, we use the case of $C_{\text{intra}}/C_{\text{inter}} = 1/10$ in the experiments. Figure 15 shows the average registration cost ratio between our approach and DHLR under various numbers of RAs (M) in the access network with different numbers of RA crossings, where $N = 1, 10, 40, 100$. According to the user locality, if the access network size becomes large, the MS becomes mobile mostly in one access network. Our average registration cost is lower than that of DHLR for all values of N.

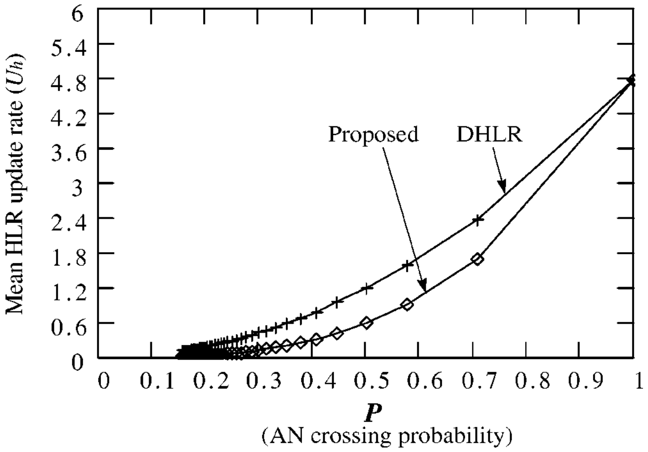

Figure 13. Mean HLR update rate under various P.

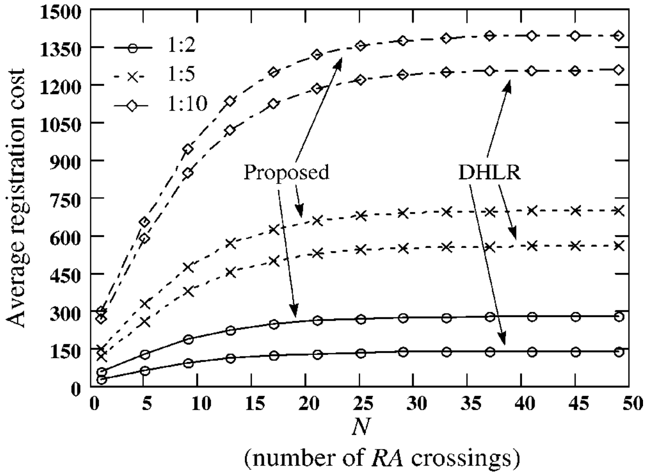

Figure 14. Average registration cost under various N.

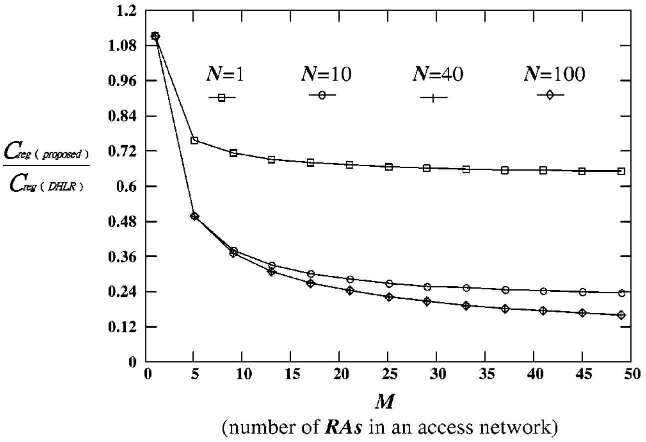

Figure 15. Average registration cost ratio under various M.

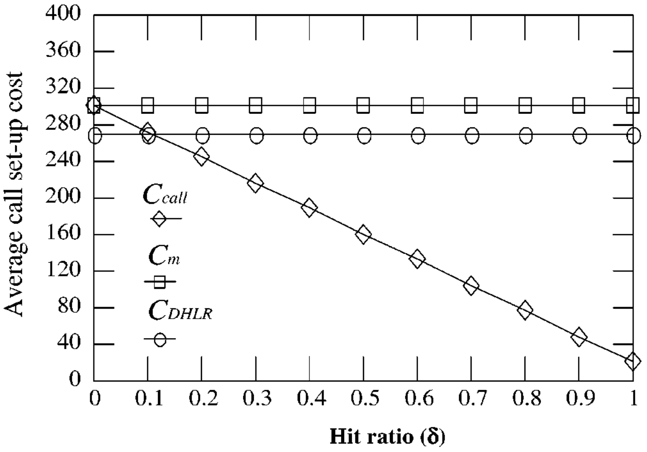

Figure 16. Average call set-up cost under different hit ratios.

In figure 16, we show the average call set-up cost between our scheme (hit and miss cases) and DHLR. The average call set-up cost of DHLR is independent of the hit ratio and we see that our hit ratio lower bound is at the hit ratio of 0.1 in comparison with the DHLR. This means that our call set-up procedure outperforms the DHLR for up to 90% of miss cases. Figure 16 also shows that the cost of adding TLCs to the network by comparing the miss case and the DHLR. Based on the definition of *CMR*, as *CMR* increases, an MS receives calls more often relative to the rate at which it changes access networks. This parameter attempts to reflect the user locality for calls received by MSs inside an access network. The relation between *CMR* and the average call set-up cost ratio is shown in figure 17. The ratio $C_{\text{call(proposed)}}/C_{\text{call(DHLR)}}$ represents the percentage of cost reduction between our call set-up with TLCs and the DHLR. This figure indicates that by using TLCs, our scheme has significantly better performance than the DHLR if *CMR* is large. For example, when $\mathit{CMR} = 5$, the call set-up cost of ours is about 75% less than that of DHLR.

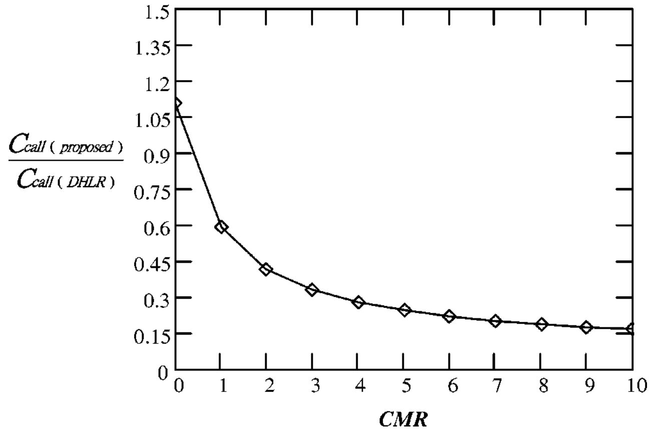

Figure 17. Average call set-up cost ratio versus *CMR*.

6. Conclusions

In our network model, we propose and integrate two mechanisms: distributed TLCs and distributed HLRs, to handle user location procedures. The location management procedures include MS registration and call set-up. These procedures fully utilize the user locality property. We use the TLC to reduce network signaling traffic and to decrease database access delay during MS registration and call set-up. If an MS receives calls or moves under user locality, our location management protocols with TLCs can avoid the shortcomings, mentioned above, of existing approaches. Other advantages of our location management procedures are as follows:

(1) Intra-AN registration is faster and more effective.
(2) Local call set-up can be established faster.
(3) In a remote call set-up, the call connection is established directly through the TLC.

Experimental results show that our registration procedure has a small mean HLR update rate and low registration cost. In addition, with a high TLC hit ratio, the call set-up cost can be reduced significantly. By the call to mobility ratio, our call set-up protocols are suitable for MSs with high called rates.

Acknowledgements

This work was supported in part by the National Science Council, ROC, under Grant NSC87-2213-E-009-030. We would like to thank the anonymous reviewers for their constructive suggestions and comments. Thanks are due to Junming Liao for helping us to prepare the revised paper.

References

[1] I.F. Akyildiz and S.M. Joseph, On location management for personal communications networks, IEEE Communications Magazine (September 1996) 138–145.

[2] B.R. Bandrinath, T. Imielinski and A. Virmani, Location strategies for personal communication network, in: *Proc. Workshop on Networking of Personal Communication Applications* (December 1992).

[3] F.V. Baumann and I.G. Niemegeers, An evaluation of location management procedures, in: *Proc. 3rd Annual International Conference on Universal Personal Communication* (September 1994) pp. 359–364.

[4] C. Cho and L.F. Marshall, An efficient location and routing scheme for mobile computing environments, IEEE Journal on Selected Area in Communications 13(5) (June 1995) 868–879.

[5] R. Cohen, B. Patel and A. Segall, Handover in a micro-cell packet switched mobile network, ACM–Baltzer Wireless Networks 2(1) (March 1996) 13–25.

[6] EIA/TIA IS-41 (Revision B), Cellular radio telecommunications intersystem operations, Technical Report EIA/TIA (July 1991).

[7] H. Harjono, R. Jain and S. Mohan, Analysis and simulation of a cache-based auxiliary user location strategy for PCS, in: *1994 Networks for Personal Communications Conference Proceedings* (1994) pp. 1–5.

[8] R. Jain, Y.-B. Lin and S. Mohan, A caching strategy to reduce network impacts of PCS, IEEE Journal on Selected Areas in Communications 12(8) (October 1994) 1434–1444.

[9] R. Jani, Y.-B. Lin, C. Lo and S. Mohan, A forwarding strategy to reduce network impacts of PCS, in: *IEEE INFOCOM '95*, Vol. 2 (April 1995) pp. 481–489.

[10] B.C. Kim, J.S. Choi and C.K. Un, A new distributed location management algorithm for broadband personal communication networks, IEEE Transactions on Vehicular Technology 44 (August 1995) 516–524.

[11] L. Kleinrock, *Queueing Systems*, Vol. 1: *Theory* (Wiley Interscience, New York, 1975).

[12] K. Kohiyama, T. Hattori, H. Sekiguchi and R. Kawasaki, Advanced personal communication system, in: *Proceedings IEEE VTC '90* (May 1990) pp. 161–166.

[13] E. Lycksel, GSM system overview, Swedish Telecommunication Admin. (January 1992).

[14] A.R. Modarressi and R.A. Skoog, Signaling system 7: A tutorial, IEEE Communications Magazine 28(7) (July 1990) 19–35.

[15] M. Mouly and M.B. Pautet, The GSM system for mobile communications, Palaiseau, France (1992).

[16] J.Z. Wang, A fully distributed location registration strategy for universal personal communication systems, IEEE Journal on Selected Areas in Communications 11(6) (August 1993) 856–860.

Kuochen Wang received the B.S. degree in control engineering from National Chiao Tung University, Taiwan, in 1978, the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona in 1986 and 1991, respectively. He is currently an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University. From 1980 to 1984, he worked on network management, and the design and implementation of Toll Trunk Information System at the Directorate General of Telecommunications in Taiwan. He served in the army as a second lieutenant communication platoon leader from 1978 to 1980. His research interests include computer networks, mobile computing and wireless networks, fault-tolerant computing, and computer-aided VLSI design.
E-mail: kwang@cis.nctu.edu.tw

Jung Huey received the B.S. degree in computer engineering from Tamkang University, Taiwan, in 1995 and the M.S. degree in computer and information science from National Chiao Tung University in 1997. His research interests include mobile computing, fault-tolerant computing, and telecommunication networks.