On Dynamically Adapting Registration Areas to User Mobility Patterns in PCS Networks

Georgios Varsamopoulos and Sandeep K. S. Gupta

Department of Computer Science
Colorado State University
Ft. Collins, CO, USA
E-mail: {varsamop, gupta}@cs.colostate.edu

Abstract

Location management is an essential service in mobile networks. It provides mechanisms for recording and querying location of mobile units in the network. This is needed for establishing calls to mobile units. In PCS networks, location management protocols such as IS-41 and GSM use statically defined Registration Areas (RAs) with two-level hierarchy of location databases. In this paper, we propose an extension to PCS location management protocol by introducing the concept of dynamically overlapping registration areas. We show through simulation that by dynamically adapting the registration areas to aggregate mobility pattern of the mobile units, we can greatly reduce the number of location updates by mobiles. Further, the cost of adapting the registration areas is shown to be low in terms of memory and communication requirements.

Keywords*: Location Management, PCS networks, Registration Areas.*

1. Introduction

Personal communication services (PCS) allow mobile users with wireless terminals to receive calls irrespective of their location in a seamless manner. PCS networks have a cellular architecture: the geographical area is divided into cells with one base station per cell. The mobile user's portable terminals communicate via wireless with fixed radio ports in the base station. Delivering calls to the mobile terminals requires that the current location (point of attachment) of the mobile terminal be known in order for the network to route the calls to the mobile terminal. The task of tracking the location of mobile terminals is known as *location management* (or *location tracking*). Location management involves two basic operations: update and search. A mobile terminal updates its location periodically or otherwise. Whenever a call needs to be delivered to the mobile, the network uses the last know location of the mobile terminal to search for the mobile in the vicinity of that area. This may involve paging for the mobile terminal in certain neighborhood of the last known location of the mobile terminal. As has been demonstrated in several works, there is a tradeoff between update and search costs. Strategy which tries to reduce one cost tend to increase the other, and vice versa. For example, if a mobile updates it's location more often then it can be searched more easily. This reduces the search cost while increasing the update cost.

A common technique used in conventional systems to balance the cost of update and search is the use of *registration area* (RA) approach to location tracking. The geographical area is divided into several registration areas, where each registration area consists of several cells. The system tracks a mobile terminal's registration area instead of its cell. Whenever a mobile terminal crosses from one registration area to another it informs its new location to the system. To setup a call to a mobile terminal, the system pages all the cells in the registration area to find the current cell of the mobile terminal. A database called *location register* (LR) is associated with each registration area to keep information about the mobiles currently registered in that registration area. There are two common standards for location management: IS-41 [3] and GSM [6]. IS-41 is used in North America and GSM is used in Europe. Both these standards use registration areas along with a two level hierarchy of location registers. The hierarchy consists of home location registers (HLR) and visitor location registers (VLR's). Location information of a mobile node is kept at both its current location registrar, i.e. its VLR, and its permanent home location registrar. This facilitates locating non local mobiles during call delivery: the home location registrar of the mobile unit is contacted to find its current location. Hence, the two level hierarchy scheme increases the location update cost while reducing the location search/call delivery cost.

In this paper we deal with the following problem. Statically defined registration areas cannot cover the entire movement pattern of the mobile users, sometimes, not even a good portion of it. This results in frequent inter-RA moves (location updates), which are costly. This is especially true when the mobile is far from its home RA, since this may require communication across the entire network. However, if we try to use lesser number of RAs, each covering wider geographical, then the load on each location registrar increases. Hence, this is not an option for reducing frequent location updates. Overlapped registration area has recently been used to address this problem [4]. In particular, with overlapping we can achieve the following: (a) increase the area handled by each LR, effectively reducing the number of inter-RA handoffs, and (b) keep the number of users (mobile hosts) handled by each LR same as in the case of no overlapping. In this paper we use a scheme that dynamically adapts to the mobility patterns and determines the area that should be covered by each LR, thus keeping the signaling overhead low. We enhance the overlapping scheme further to allows for dynamic overlapping of the registration areas.

The scheme proposed in this paper keeps track of aggregate user mobility and uses this information to decide the degree of overlapping between neighboring registration areas which will reduce the number of registration area handoffs while keeping the call delivery cost down. The decision to adjust the registration area is done in a distributed manner and the protocol ensures that if the mobility pattern are stable then the registration areas would also stabilize. The memory and communication overhead of the proposed scheme is low making the scheme scalable. We have simulated our protocol for various mobility patterns. Our simulation results show that the idea of overlapping is good when there is locality in the overall movement of the mobile users. However, this scheme provides little improvement when it is applied in highway-like movement patterns.

In the rest of the paper we first discuss some related work in the literature. We then present our system model in Section 3. Our proposed algorithm for dynamically adjusting the registration areas is described in Section 4. In Section 5, we present our simulation model and results. Finally, Section 6 concludes the paper.

2. Related Work

In this section we give a brief description of some location management schemes in the literature.

Bejerano and Cidon [1] have presented a location management scheme which is based on the assumption that the communication cost between two RAs is a constant factor of the geographical distance between the location registrars of the two RAs. They define a logarithmic number of levels of RAs. Each RA has a radius twice as much as the ones at the radius of RA level right below it. The RAs at each level overlap as much as their radius. This scheme has a good average behavior, but it has a substantial delay in the worst case. It also has a storage space drawback, because it uses a logarithmic number of profile replicates per user.

Ho and Akyildiz [4] have introduced a three level database hierarchy with the number of number levels dynamically adapting to the user pattern (call to mobility ratio - CMR). They have also introduced the idea of registration area overlapping. Nevertheless, they do not describe how to determine the degree of overlapping between adjacent RAs.

Kryukova, Massingill, and Sanders [5] have proposed an integrated scheme of location management and call routing. Their scheme is based on an acyclic undirected graph on which they maintain routing pointers. Rajagopalan and Badrinath [8] have proposed a hybrid scheme based on the Mobile-IP architecture that uses either the home-agent/visitor-agent paradigm or the direct location update between parties that communicate frequently. Das and Sen [2] have proposed a scheme for predicting the location of a user based on its last known location and velocity. Prakash and Singhal [7] have proposed a user profile replication scheme. Their idea is to have the information of location replicated among quorums of location servers organized into a coterie.

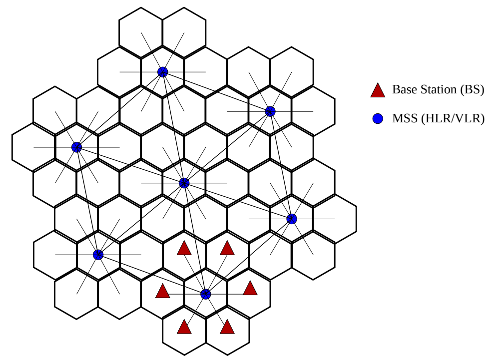

Figure 1. The System Model (Cells are organized into clusters (here in clusters of 7) forming the Registration Areas (RAs). The MSSs (LRs), located at the center of each RA, are connected through a hexagonal mesh.)

3. System Model

In this section, we describe the system model used in this paper. As shown in Figure 1, a cellular communication network consists of an array of hexagonal communication cells. We assume that there are $\mathcal{N}$ cells in the system numbered from 1 to $\mathcal{N}$. Each cell except the boundary cells have six neighbors. A base station (BS) is in-charge of a cell. Every base station is connected to a Mobile Switching Station (MSS). The MSS are connected to an interconnection network; hence each MSS can communicate with other MSSs in the system using the wired backbone. Each MSS is in charge of providing PCS service to a set of cells called its Registration Area (RA). We assume that, initially, the RA of a MSS consists of all the cells whose BS are directly connected to the MSS. We call these cells as **core cells** of that MSS. Associated with each MSS are two location databases: Visitor Location Registrar (VLR) and Home Location Registrar (HLR). A VLR of a MSS keeps information about all the mobile hosts (MHs) in its registration area. To facilitate search of a MH, each MH is statically associated with a Home Location Registrar (HLR), which keeps information about the current location of the MH, i.e. the id of the MSS in whose RA the MH is currently residing. Hence, a location registrar maps an MH id to an MSS id.

When an MH which is actively involved in communication moves from one cell, say c_{old}, to another cell, say c_{new}, a *handoff* takes place, after which the MH receives and sends all its communication via base station of c_{new} instead of base station of c_{old}. This involves signaling between MH, BS, and MSS and resource allocation and deallocation. Further, if the two cells c_{old} and c_{new} belong to different registration areas then the HLR of the MH is informed of the new location of the MH. Further, the VLR of MH's old registration area and its new registration area also need to be updated, i.e. the MH's record in the old RA's VLR is deleted and a record for the MH in the VLR of its new RA is added. Hence, **inter-RA handoffs** are more expensive than **intra-RA** handoffs.

4. Proposed Scheme

In the proposed location management scheme, associated with each MSS is a RA. Since there is one-to-one correspondence between the MSSs and the RAs, we will refer to the RA associated with MSS k as RA k. The RA associated with an MSS can be viewed as a set of cells. As opposed to static schemes, the membership of this set is dynamic in our scheme. Periodically, every MSS decides which cell will be included in or excluded from its RA. We require the dynamic RAs to satisfy the following properties:

Property 1 *An RA has at least one cell.*

Property 2 *The subgraph of the cell adjacency graph*[1] G *induced by any RA is connected.*

Property 1 guarantees that no RA is empty and Property 2 guarantees that no RA has any cell or a group of cells which is disconnected from remaining cells in the RA. The (ordered) set of all RAs in the system at any instance of time is referred to as a *configuration* of the system, i.e., a configuration $C = \{RA_1, RA_2, \ldots, RA_M\}$, where RA_i, $1 \leq i \leq M$, denotes RA i. Further, changing the configuration of the system is referred to as *reconfiguration*. The MSSs in the system reconfigure the system in a distributed manner at a fixed interval of time T.

Initially, the RA k consists of only those cells whose BSs are directly connected to MSS k. These cells are referred to as **core** cells of RA k and will be denoted as $Core(k)$. The significance of the core cells of an RA is that an MSS never excludes any core cell from its RA. Hence, the notion of core cells guarantees Property 1 is always satisfied. Further, it ensures that each MSS has certain minimal load. In any configuration, we define two types of handoffs.

- an **intra-RA handoff** takes place whenever an MH which is registered with RA k moves from a cell c (which is in RA k) to another cell d which is in RA k, and
- an **inter-RA handoff** takes place whenever an MH which is registered with RA k moves from a cell c (which is in RA k) to another cell d which is not in RA k. *After the handoff the MH is registered with the RA l such that $d \in Core(l)$.*

Further, there are two types of call-deliveries:

- an **intra-RA call-delivery** is one in which both the caller MH and the callee MH are in the same RA.
- an **inter-RA call-delivery** is one in which the caller and the callee MHs are in different RAs.

4.1. Registration Area Overlapping

The main advantage of overlapping the RAs is that each RA can provide service to more MHs within their covered area; reducing the number of inter-RA handoffs and consequently the overhead to update the MH's HLRs. On the other hand, the drawback of overlapping the RAs is that the communication overhead for call delivery and location registration within the RA is increased due to increase in the diameter of the RA. The increase in the overhead depends upon the underlying network topology. If this overhead is ignored, then we can reach the extreme configuration where every RA covers the entire network area, so that no communication to the HLRs is required.

[1]The cell adjacency graph $G = (V, E)$ has V as the set of all cells in the network; and an edge $(c, d) \in E$ iff c and d are neighboring cells.

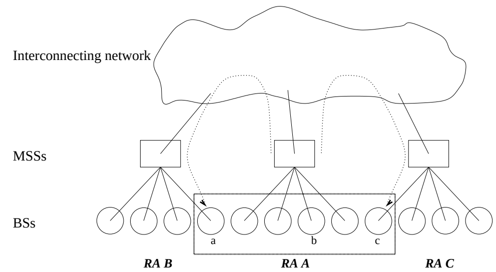

Figure 2. The increase of signaling cost is shown here. Even if MH's a and b belong to the same RA, any signaling between them should go through two MSSs.

In Figure 2 we show how the increase in signaling cost happens. The core of the RA of the MSS consists of all the cells whose BS are directly linked to the MSS. In Figure 2, RA A has been expanded to fit one cell from each of its neighboring RAs: Assume that a and b are MHs that are registered with RA A. If there is a connection to be established between MH a and b, the signaling cost will be almost as if they were in separate RAs, i.e. the signaling has to go through the two MSSs. In general there are following cases for intra-RA and inter-RA handoffs:

Case 1: MH moves from a core cell to another core cell.

Case 2: MH moves from a core cell to another non-core cell.

Case 3: MH moves from a non-core cell to another non-core cell (this case is not possible for intra-RA handoffs).

The main difference between an inter-RA and an intra-RA handoff is that; (i) there are two LRs that access their database in the inter-RA handoff compared to only one in the case of intra-RA, and (ii) there are 2 extra backbone messages that are transmitted, usually between two neighboring MSSs. Figure 3 summarizes the signaling costs in terms of: i) t_{cv}: communication cost between a BS and its MSS; ii) t_{vv}: communication cost between neighboring MSSs; iii) t_{bb}: average communication cost between two arbitrary MSSs, i.e. $t_{bb} =$ (avg. nos. of hops between two MSSs)t_{vv} ; and iv) t_a: cost to access a location registrar. Note that in our scheme it is not possible to have a non-core to non-core inter-RA handoffs since after an inter-RA handoff the new RA to which a mobile is registered to is the default RA of its new cell i.e. the new cell is a core cell of mobile host's new RA.

Similarly, there are three cases for both intra-RA and inter-RA call-deliveries:

Case 1: Both MHs belong to the core of an RA

Case 2: One MH belongs to the core and the other belongs to the expanded part of the RA.

Case 3: Both MHs belong to the expanded part of RA.

The costs for various cases are shown in Figure 3.

Case #	Intra-RA handoff	Inter-RA handoff
	Intra-RA call-delivery	Inter-RA call-delivery
1	$4t_{cv} + t_a$	$4t_{cv} + 2t_a + 2t_{bb}$
	$4t_{cv} + t_a$	$4t_{cv} + 2t_a + 2t_{bb}$
2	$4t_{cv} + t_a + 2t_{vv}$	$4t_{cv} + 2t_a + 2t_{vv} + 2t_{bb}$
	$4t_{cv} + t_a + 2t_{vv}$	$4t_{cv} + 2t_a + 2t_{vv} + 2t_{bb}$
3	$4t_{cv} + t_a + 4t_{vv}$	N/A
	$4t_{cv} + t_a + 4t_{vv}$	$4t_{cv} + 2t_a + 4t_{vv} + 2t_{bb}$

Figure 3. Costs for different handoffs/call-deliveries.

4.2. Dynamic Overlapping

The effect of increasing the size of an RA is that it reduces the number of inter-RA handoffs. The advantage of overlapping RAs is that it can further reduce the number of inter-RA handoffs; however, intra-RA signaling cost may increase due to overlapping RAs. We need to find the optimal degree of overlapping so that the gain of the reduction in inter-RA moves is not over-weighed by the loss due to increased intra-RA signaling cost.

This paper is proposing a dynamic (adaptive) reconfiguration of the RA in terms of degree of overlapping. This means that we let the network decide the optimal degree of overlapping depending upon the current aggregate mobility pattern. The idea is simple: if there is a high frequency of hand-offs from RA A to RA B then RA A tries to include the cells of RA B where the users are moving in. This is done only when the new cell to be included in the RA A is not far (that is the communication between the new cell and the RA server is not expensive). On the other hand, when the hand-offs between two cells are rare. There is no point in keeping that cell within the range of a RA, so it is excluded from the RA's cells. The use of RA shrinkage is to prevent irreversible expansion of RAs. In this way, the RAs can adapt to the (changing) mobility patterns of the users and find a good partitioning scheme themselves for each occasion.

In order to facilitate orderly growth and shrinking of RAs, an MSS only includes or excludes cells from its RA's current boundary. The algorithm uses the following two types of boundaries:

Definition 1 *A cell $c \in RA$ is in the internal boundary of that RA iff $\exists d \notin RA$ such that cell d is a neighbor of cell c. The internal boundary of RA k will be denoted as $I\_Boundary(k)$.*

Definition 2 *A cell $d \notin RA$ is in the external boundary of that RA iff $\exists c \in RA$ such that cell d is a neighbor of cell c. The external boundary of RA k will be denoted as $E\_Boundary(k)$.*

All the cells in $I\_Boundary(k)$ are *candidate cells for exclusion* from the RA k and all the cells in $E\_Boundary(k)$ are *candidate cells for inclusion* into RA k. However, since core cells cannot be deleted from an RA, only cells in $I\_Boundary(k) - Core(k)$ can be excluded from an RA. The decision to include or exclude a candidate cell is based on whether the resulting configuration will have a lower overall system load. For a given system configuration C, mobility pattern $\mathcal{M}$, and call pattern $\mathcal{C}$, we define system load, $SystemLoad(C, \mathcal{M}, \mathcal{C})$ as the combined signaling load (in terms of message time complexity) as a result of all the handoffs due to $\mathcal{M}$ and call-deliveries due to $\mathcal{C}$. For the purpose of making cell inclusion-exclusion decisions, we partition the entire system load into loads per MSS, i.e.

$$SystemLoad(C, \mathcal{M}, \mathcal{C}) = \sum_{k \in \mathcal{R}} Load(k, \mathcal{M}, \mathcal{C}),$$

where $\mathcal{R}$ is the set of all MSSs and $Load(k, \mathcal{M}, \mathcal{C})$ is the signaling load attributed to MSS k. We note here that in case of inter_RA handoffs and call-deliveries we split the signaling overhead equally between the two MSSs involved. Further, we will simply use $Load(k)$ to denote the load of MSS k.

4.3. Algorithm Description

The idea behind the algorithm is to decide whether to include or exclude a candidate cell. At intervals of time T, called *reconfiguration period*, each MSS k:

1. Computes the following two costs for each candidate cell $i \in (I\_Boundary(k) - Core(k)) \cup E\_Boundary(k)$:
   (a) $Cost_{in}(k, i)$: the portion of $Load(k)$ due to the cost of performing call deliveries and hand-offs for users in the candidate cell i, if cell i is included in the RA k.
   (b) $Cost_{ex}(k, i)$: the portion of $Load(k)$ due to the cost of performing call deliveries and hand-offs with users in candidate cell i, if cell i is not included in the RA k.
2. Excludes all cell $i \in I\_Boundary(k) - Core(k)$ from RA k if $Cost_{in}(k, i) > Cost_{ex}(k, i)$.
3. Removes all cell j from $E\_Boundary(k)$ which are neighboring cell to any cell i removed in Step 2.
4. Includes all cell $i \in E\_Boundary(k)$ in RA k if $Cost_{in}(k, i) < Cost_{ex}(k, i)$.

An MSS tries to shrink its RA (Step 2) before it tries to grow (Step 4). Step 3 ensures that there are no "holes" in a RA.

4.4. Cost Computation

When a cell i is not serviced by MSS k, the interaction between cell i and RA k consists of inter-RA call deliveries and inter-RA hand-offs between cells of RA k and cell i. On the other hand, when the cell i is serviced by RA k, the interaction between cell i and RA k consists of inter-RA call deliveries and hand-offs between cell i and all other RAs in the system, intra-RA call deliveries and hand-offs between cell i and all the remaining cells of RA k. Analyzing the two costs we have:

$Cost_{in}(k, i)$ = Cost for searching the database in RA k and signaling the cell i upon call requests + Cost for performing intra-RA hand-offs for the users registered with RA k in cell i + Cost for performing inter-RA hand-offs for the users registered with RA k in cell i.

$Cost_{ex}(k, i)$ = Cost for performing inter-RA hand-offs for users registered with RA k that move to cell i + Cost for performing inter-RA call deliveries between users in RA k and those is cell i which would have been registered with RA k.

In order to be able to compute those values, the MSS has to keep information about the cell i for both the case the cell is included and the case the cell is excluded from the RA. There are basically three factors that affect the cost values:

- $R_h(k,i)$: The number of incoming and outgoing handoffs between cell i and the cells of RA k in last reconfiguration period.
- $R_{hout}(k,i)$: The number of handoffs done by users in i that are registered (or would be registered) to RA k and move to a cell out off the RA k in the last reconfiguration period.
- $R_c(k,i)$: The number of incoming and outgoing calls between cell i and users in RA k in the last reconfiguration period.

We define the following notations for the give configuration C, mobility pattern $\mathcal{M}$ and call pattern $\mathcal{C}$:

- $n_{moves}(k,i,m,j)$: is the total number of moves by MHs registered with RA k in cell i to cell j of RA m. Note that if cell $i \notin$ RA k then $n_{moves}(k,i,m,j)$ is the number of users which arrived from cell i of RA k to cell j of RA m.
- $n_{calls}(k,i,m,j)$: is the total number of calls from MHs registered with RA k in cell i to MHs registered with RA m in cell j. If cell i is currently not included in RA k, $n_{calls}(k,i,m,j)$ is the number of calls made/received by users which arrived from cell i of RA k to/from user in cell j of RA m.

A BS for cell i updates the relevant $n_{moves}(*,i,*,*)$ and $n_{moves}(*,*,*,i)$ counters on each handoff it is involved in. Similarly, it updates the relevant $n_{calls}(*,i,*,*)$ and $n_{calls}(*,*,*,i)$ counters on each call-delivery it is involved in. In terms of these counters, $R_h(k,a)$, $R_{hout}(k,a)$, and $R_c(k,a)$ can be expressed as follows:

$$\begin{aligned}
R_h(k,a) &= \sum_{i \in RA\ k} n_{moves}(k,i,k,a) + \\
&\quad \sum_{i \in RA\ k} n_{moves}(k,a,k,i) \\
R_{hout}(k,a) &= \sum_{m \in \mathcal{R}-\{k\}} \sum_{j \in RA\ m} n_{moves}(k,a,m,j) \\
R_c(k,a) &= \sum_{m \in \mathcal{R}} \sum_{j \in RA\ m} n_{calls}(k,a,m,j) + \\
&\quad \sum_{m \in \mathcal{R}} \sum_{j \in RA\ m} n_{calls}(m,j,k,a)
\end{aligned}$$

Using the above R-quantities, MSS k can compute the cost values as follows:

- $Cost_{in}(k,i) = \alpha R_h(k,i) + \beta R_c(k,i) + \gamma R_{hout}(k,i)$,
- $Cost_{ex}(k,i) = \gamma R_h(k,i) + \delta R_c(k,i)$

where α, β, γ, and δ are defined as follows:

- α: Signaling cost for performing an intra-RA hand-off for cell i under RA k.
- β: Signaling cost for performing a call delivery from/to cell i from LR k when i is included in k.

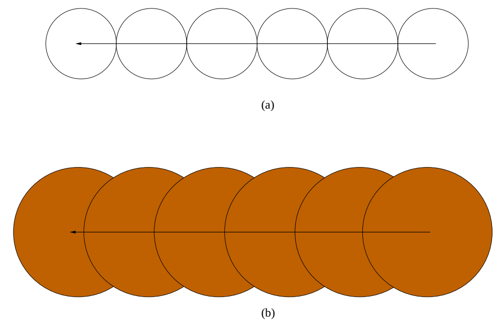

Figure 4. Overlapping doesn't help in highway cellular services. For the first figure (a), we have 5 handoffs, which are as many as in (b).

- γ: Signaling cost for performing an inter-RA hand-off for cell i under RA k.
- δ: Signaling cost for performing an call delivery from/to cell i from RA k when i is not included in k.

We assign α (γ) to be the average costs for the three intra-RA (two inter-RA) handoff costs shown in Table 3. Similarly, we assign β (δ) to be the average costs for the three intra-RA (inter-RA) call-delivery costs shown in Table 3.

In summary our dynamic overlapping scheme, each MSS periodically decides whether to include or exclude cells from its RA based on the information collected by BS in the last reconfiguration period. However, this incurs only a minor overhead since the computation involves only a few simple arithmetic and logic operations and few message exchanges between a MSS and its BSs.

5. Simulation

In this section, we first describe the user mobility models used in the simulation and then present some simulation results. We model three types of movement patterns:

1. **highway movement**: In this pattern, we have a large linear array of cells. We organize consecutive cells into registration areas. The movement is monotonic in the sense that the users move either from left to right or right to left (see Figure 4).
2. **1-dimensional random walk** In this mobility model, the movement is no longer monotonic, but stochastic, according to the following transition matrix P:

$$P = \begin{bmatrix}
\cdots & & & & & & & & \\
& \frac{1}{16} & 0 & \frac{15}{16} & & & & & \\
& & \frac{1}{9} & 0 & \frac{8}{9} & & & & \\
& & & \frac{1}{2} & 0 & \frac{1}{2} & & & \\
& & & & \frac{8}{9} & 0 & \frac{1}{9} & & \\
& & & & & \frac{15}{16} & 0 & \frac{1}{16} & \\
& & & & & & & & \cdots
\end{bmatrix},$$

where P_{ij} is the probability that an MH in cell i will move to cell j in the next step. We simulate a linear array of 400 cells divided in nine RAs. We evenly distributed 10000 MHs in the cells and ran the simulation for different values of call frequency (from 1 call per 15 mins. per user to 1 call per hour

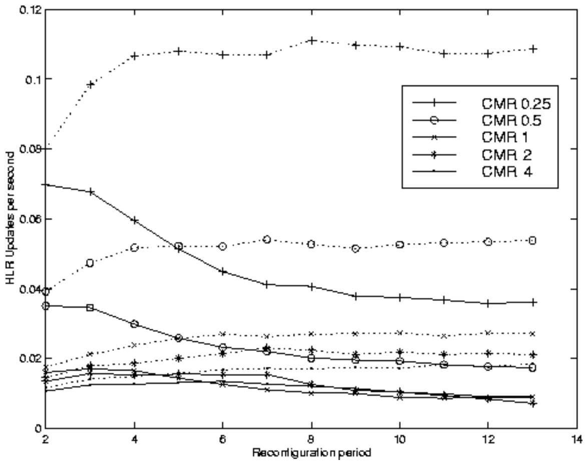

Figure 5. Results from the 1-dimensional model for various values of CMR: The dotted lines show the update rates at the HLRs, when no overlapping is used. Solid lines show the update rates at the HLRs, when overlapping is used.

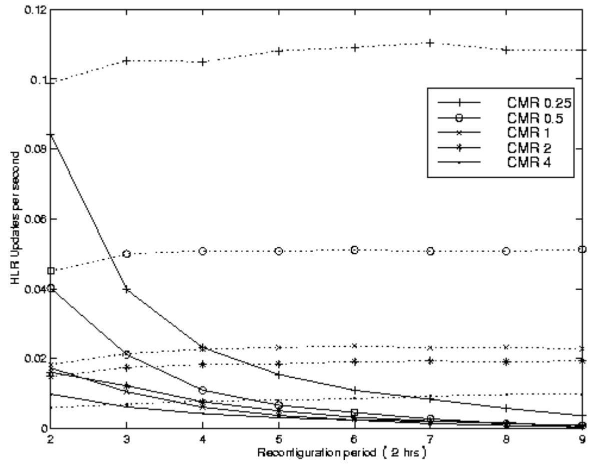

Figure 6. Results from the 2-dimensional model for various values of CMR: The dotted lines show the update rates at the HLRs, when no overlapping is used. Solid lines show the update rates at the HLRs, when overlapping is used.

per user) and mobility frequency (from 1 handoff per 15 mins. per user to 1 handoff per hour per user).

3. **2-dimensional random walk**: In this model, we have a large 2-dimensional array of cells. The mobility pattern is a random walk with a separate *pole of attraction* for each mobile station, the probability a mobile station moves further from its pole is inversely proportional to the square of the current distance from its pole. There are 10,000 MHs that are move around in a grid of 50x50 cells. The grid of cells is organized into a grid of 5x5 RAs, i.e. each RA has 25 cells. Initially, each MH is assigned their basic cell. When the simulation starts, the MHs start moving, and the RAs start expanding, according to the data they collect. The mobility of the MHs is limited by a quadratically decreasing probability function, so that they rarely go beyond 5 cells from their initial cell. On each move of a MH, though, there is a small probability that they jump to any cell in the network. Although the probability is low, there are about 200 MH's at each step on average that are assigned a random current cell. This gives a spontaneous and occasional movement of the MHs beyond their localized movement pattern. The idea is to test whether the algorithm can distinguish between this spontaneous movement and the normal periodic movement and reach equilibrium.

Our simulation results have shown that overlapping (either static or dynamic) doesn't help in highways (see Figure 4 for justification). In the case of 1-dimensional and 2-dimensional random walk model our results have shown that overlapping really helps. Figure 5 and Figure 6 show the HLR update rates versus reconfiguration periods for 1-d and 2-d random walk mobility pattern, respectively. Each figure has plots for various CMR (Call-to-Mobility Ratio). It can be seen that for a given CMR (Call-to-Mobility Ratio) the dynamic scheme has lower handoff rates than the static scheme.

6. Conclusions

We have proposed a scheme for dynamically adjusting the registration areas in a PCS network. Each MSS in the system adjusts its RA periodically based on the data regarding handoffs and call-deliveries collected during last reconfiguration period. Simulation results show that our dynamic overlapping scheme adapts to user mobility pattern and results in lower handoff rates than the static registration area scheme. Future work includes incorporating dynamic overlapping scheme in other hierarchical location management scheme and studying their performance.

References

[1] Y. Bejerano and I. Cidon. An Efficient Mobility Management Strategy for Personal Communication Systems. pages 215–222, April 1998.

[2] S. K. Das and S. K. Sen. Adaptive Location Prediction Strategies Based on a Hierarchical Network Model in Cellular Mobile Environment. Proceedings of Second International Mobile Computing Conference (IMCC), March 1996.

[3] EIA/TIA. Cellular radio-telecommunications intersystem operations, is-41 revision b. Technical report, EIA/TIA, 1991.

[4] J. S. M. Ho and I. F. Akyildiz. Dynamic Hierarchical Location Management in PCS Networks. *IEEE/ACM Transactions on Networking*, 5(5):646–660, October 1997.

[5] S. Kryukova, B. Massingill, and B. Sanders. An Algorithm for Distributed Location Management in Networks of Mobile Computers. Technical report, California Institute of Technology, Computer Science department, 1996.

[6] M. Mouly and M. B. Pautet. The GSM system for mobile communications. Technical report, 49 rue Louise Bruneau, Palaiseau, France, 1992.

[7] R. Prakash and M. Singhal. Dynamic Hashing + Quorum = Efficient Location Management for Mobile Computing Systems. In *ACM Symp. Principles Distributed Computing*, page 291, August 1997.

[8] S. Rajagopalan and B. R. Badrinath. An Adaptive Location Management Strategy for Mobile IP. MobiCom, 1995.