C. 2

State University of New York at Stony Brook
College of Engineering and Applied Sciences

Technical Report No. 775

A Novel Location Management Scheme for Personal Communications

by

Daqing Gu and Stephen S. Rappaport

Department of Electrical and Computer Engineering
State University of New York
Stony Brook, New York 11794-2350

e-mail: dgu@sbee.sunysb.edu, rappaport@sunysb.edu

Date: June 28, 1999

C. 2

State University of New York at Stony Brook
College of Engineering and Applied Sciences

Technical Report No. 775

A Novel Location Management Scheme for Personal Communications

by

Daqing Gu and Stephen S. Rappaport

Department of Electrical and Computer Engineering
State University of New York
Stony Brook, New York 11794-2350

e-mail: dgu@sbee.sunysb.edu, rappaport@sunysb.edu

Date: June 28, 1999

A Novel Location Management Scheme for Personal Communications

Daqing Gu and Stephen S. Rappaport

Electrical and Computer Engineering
State University of New York at Stony Brook
Stony Brook, NY 11794-2350

e-mail: dgu@sbee.sunysb.edu, rappaport@sunysb.edu

Abstract

We consider a novel location management scheme in which the databases are logically organized into multiple collections of two kinds of sets, *write sets* and *read sets*. These multiple collections of write sets and read sets are in turn logically arranged into multiple steps. Each collection corresponds to one step. The sizes of write sets and read sets at different steps can be different. Any write set and any read set at the same step intersect. The scheme accounts for mobility and call arrival rate of mobile users and dynamically changes the step of write sets and read sets to which a mobile user registers. Mobile users having low mobility and high call arrival rate will register at a step of large-size write sets and small-size read sets, while mobile users with high mobility and low call arrival rate will register at a step having small-size write sets and large-size read sets. Upon a location update, a mobile user's location information is written into a write set of its registered step. For each call arrival, the called mobile user's location information can be retrieved from a read set at a certain step. An analytical model is developed to evaluate performance, and implementation issues are also discussed. Example performance characteristics show that substantial reduction in the average combined signaling cost for location update and location query on the backbone network can be achieved in comparison with a scheme having the same number of databases that are logically arranged as a single collection of sets (every two of which intersect).

The research reported in this paper was supported by the U.S. National Science Foundation under grant no. NCR 94-15530. General research support from Hughes Network Systems is gratefully acknowledged.

1. Introduction

In mobile cellular communications, the service coverage area is divided into many cells, each served by a base station through which mobile users communicate via wireless links. As a mobile user moves, its network access node (base station) changes. In order to set up a call connection to a mobile user, a suitable network access node must be identified. In most current cellular systems, a centralized two-level database structure, which includes two types of databases - home location register (HLR) and visitor location register (VLR), is used for location tracking [1,2]. In this scheme, the cells in the service area are grouped into location areas (LAs), each consisting of a cluster of cells and assigned a unique identifier (ID). The base stations of the cells in a group of LAs are connected by fixed (usually wireline) links to a mobile switching center (MSC), which provides switching functions and coordinates location registration and call delivery [2]. The MSCs in the system are connected to a wireline *backbone network*. In general, there is an HLR for each geographical region that may include one or more MSCs, and there is a VLR associated with each MSC. Each mobile user is permanently associated with an HLR in the geographical region where service is provided. Each VLR is used to store the current LAs of the mobile users visiting the associated MSC. The ID of the VLR that is currently serving a mobile user is stored in the user's HLR.

In order to locate a mobile user when a call arrives, each mobile terminal is required to report its LA to the system whenever it enters a new LA by sending the ID of the new LA to the nearby base station. The system updates the user's LA ID in the user's serving VLR. If the user's previous and new LAs are served by the same VLR, no further action is required. Otherwise, the system replaces the ID of the previous VLR with the ID of the new VLR in the user's HLR. The above procedure is called *location update*. The exchange of information needed to accomplish location updates involves radio and wireline signaling. Primary focus is on the wireline signaling in this paper. When an incoming call arrives, the system queries the called mobile user's HLR. The HLR forwards the query to the user's serving VLR to get the user's current LA. This process is called *location query*. Once the ID of the called mobile user's current LA is retrieved, the system pages the called mobile user in the user's current LA to determine the user's serving base station. The call connection is then set up through that base station.

Cellular communications are evolving toward a new generation that can support several hundred million subscribers with different mobility and call activities throughout our planet, while providing various types of services comparable to that of wireline users. Although the above-described HLR-VLR scheme is suitable for the current cellular communication systems, it may not be adequate for future mobile communication services. The main reasons are as follows:

1. *Excessive signaling traffic generated by location update and location query on the backbone network*: The signaling traffic transmitted between HLR and VLR on the backbone network can be excessive when mobile users are located far away from their HLR, especially if the mobile users frequently change LAs and receive calls.
2. *Home dependent numbering* [5]: In the HLR-VLR scheme, a mobile user's phone number depends on its home location (its HLR). Since mobile users in future personal communication systems may roam nationally or internationally, they would like to have the same phone number for their entire life, regardless of whether they change service providers, or change their home location.
3. *Vulnerable to the failures of HLRs*: Since a mobile user's location information is only stored in its HLR, if its HLR fails, the system can not find the mobile user's current LA.

To reduce the signaling traffic generated by location update and query on the backbone network while keeping the basic HLR-VLR database structure, a *pointer forwarding scheme* [3] and a *per-user location caching strategy* [4] have been proposed. Each can reduce the signaling overhead incurred by location update and location query under a certain conditions.

A distributed location management scheme using *quorums* to organize databases has been proposed [5], [6]. In the scheme, the databases are organized into a single collection of sets every two of which intersect. Each set in this collection is called a *quorum*, and the collection of sets with intersection property is called a *quorum system* [8], [9]. Each time a mobile user does a location update, the system stores the ID of the user's new LA in all databases of a selected quorum, and removes the ID of the user's previous LA in the quorum that was selected when the user last did its location update. When an incoming call arrives, the system queries all databases in one quorum. Because of the intersection property of a quorum system, the called mobile user's current LA can be retrieved from any quorum. In comparison with centralized schemes, this scheme is more robust in case of database failure. This is because multiple copies of a mobile user's current LA are maintained in the system. The scheme is especially suitable for home independent numbering schemes because of the distributed database structure. The work of [5] primarily focused on load balancing among location servers (databases). While, the work of [6] emphasized the trade-off between the cost of location updates and reliability of the system.

In this paper, we try to reduce the signaling traffic for location update and location query on the backbone network in the quorum-based scheme. One observation is that the signaling traffic generated by location update and location query depends on the *size* of the quorums (the number of databases in a quorum) as well as user mobility and call activity. So it is desirable to keep the size of each quorum as small as possible. We propose to organize the databases into multiple collections of two kinds of sets, *write sets* and *read sets* [7], in which any write set and any read set in the same collection intersect. A *write set* is a set of databases that are used to store the IDs of mobile users current LAs, and a *read set* is a set of databases that are used in retrieval of mobile users current LAs. In this structure the requirement is that *any* write set and *any* read set in the same collection intersect. This gives some flexibility in making write sets and read sets smaller than the quorums constructed by using one collection of one kind of set. We will illustrate this in subsequent discussion.

If a group of relatively large-size write sets is constructed, the size of each read set can be decreased while maintaining intersection between any write set and any read set in one collection. A group of relatively small-size write sets can be built by increasing the size of each read set as well. For example, if a write set is constructed that includes all databases in the system, the read sets can be made as small as one database while keeping intersection between any write set and any read set. Similarly, if a read set is constructed that includes all databases in the system, the write sets can be made as small as one database. In this way, multiple collections of write sets and read sets with different sizes can be constructed. The use of variously sized write sets and read sets allows service to the mobile users with different mobility and call activities without excessive signaling traffic for location update and query. Specifically, these multiple collections of write sets and read sets are logically arranged into multiple steps with each collection of write sets and read sets corresponding to one step. The sizes of write sets and read sets at different steps are different. Any read set and any write set at the same step intersect. Mobile users having low mobility and high call-arrival-rate will register at a step of large-size write sets and small-size read sets, while users with high mobility and low call-arrival rate will register at a step having small-size write sets and large-size read sets. At any given time, a mobile user is registered at a suitable step. The registered step is dynamically changed according to the user's mobility and call activity estimate. Upon a location update, the ID of a mobile user's new LA is written into a write set of its registered step. For each call arrival, the called mobile user's current LA can be obtained from a read set at a certain step. In this way, the signaling traffic for location update and query is reduced, while the properties of distributed database structures, such as suitability for home independent phone numbering scheme and robustness to database failures is retained.

The concept of *write sets* and *read sets* has been used to organize databases for location tracking [7]. In [7], a hierarchy of distributed regional directories is maintained, where each regional directory is based on a decomposition of the system into regions. A regional directory is a collection of sets of nodes (databases) in the region, consisting of a write set and a read set for each node, with the property that any write set and any read set of the regional directory intersect. Write sets are used to store mobile users current LAs. Read sets are used to retrieve the current LAs of mobile users. The purpose of a regional directory is to enable any terminal to directly track any mobile user residing in that region. If the current LA of a mobile user is not found in a regional directory, the search goes to a higher level that covers a relative larger region.

Our proposed scheme is different from [7] in the following ways, (1): Steps in the proposed scheme are formed according to the sizes of write sets and read sets, and a collection of write sets and read sets at any step covers the same area - the entire service area. While, levels in the scheme of [7] are formed according to the regions covered by a collection of write sets and read sets. (2): The system assigns different steps of write sets and read sets in the proposed scheme to individual mobile users with different mobility and call arrival rate. While, the sizes of write sets and read sets in [7] are the same for all mobile users. (3): The work of [7] tried to bound the signaling cost for location update and query by using the locality of mobile users movements and call origination. While, we are going to reduce the signaling cost for location update and query by assigning the different sizes of write sets and read sets for individual mobile users.

2. Construction of Write Sets and Read Sets

We first discuss *quorums* and their construction as a preliminary to demonstrate our proposed write sets and read sets construction. A quorum system is defined in [8], [9], [10]. A *set* is a collection of elements, and a *set system* is a collection of sets. A set system is said to satisfy the *intersection property* if every two sets in the set system have at least one element in common. Set systems that have the intersection property are known as *quorum systems*, and the sets in such a system are called *quorums*.

Several quorum system construction methods have been described in [5], [9]], [10]. One method is the *grid-based construction scheme*. Suppose that the total number, N, of elements is a perfect square, then $N = t^2$, where t is an integer. A $t \times t$ grid can be constructed as shown in Fig. 1. Each grid point represents an element. Each row and each column contain t elements respectively. The rows are numbered 0 to t-1 from the top to the bottom, while the columns are numbered 0 to t-1 from the left to the right. Thus, each element has a pair of indices (i,j), (i, j=0, 1, 2, … t-1) corresponding to its row and column. A quorum q_i can be constructed by including all elements on the i-th row, and one element from each of every other row. Specifically, we let quorum q_i consist of all the elements on the i-th row and the i-th column (i=0, 1, 2, … t-1). Any two distinct quorums, say q_i and q_j, have the elements (i,j) and (j,i) in common. In this way, a quorum system that contains t quorums is constructed. It is clear that each quorum contains $(2 \cdot t - 1)$ elements.

When the grid-based scheme is used to construct a quorum system for a quorum-based location management strategy, the elements in the grid are identified as databases. So each quorum contains $(2 \cdot t - 1)$ databases and there are altogether $N = t^2$ databases. For each location update (or location query), the system has to update (or query) all the databases in a quorum, i.e., $(2 \cdot t - 1)$ databases.

From Fig. 1, we observe that these $N = t^2$ databases can also be logically arranged in two groups of sets. Each set of the first group consists of all databases in a row, while each set of the second group consists of all databases in a column. Thus, any set of the first group intersects all sets of the second, and vice versa. We elect one of these two groups of sets as write sets in the proposed scheme, and the other group as read sets. Any read set and any write set intersect. If the ID of a mobile user's current LA is stored in one write set, it can be obtained from any read set. Since the number of databases in each write set or read set is *t*, only *t* databases have to be accessed for each location update (or query). The number of databases in a read or write set is called its *size*. Compared with $(2 \cdot t - 1)$ databases in a quorum in quorum-based scheme, this implies that the proposed scheme allows almost 50% savings for large *t* in terms of the number of databases accessed per location update or query. Additionally, the sizes of write sets and read sets can be adjusted by changing the square grid to a rectangular grid with $N = r \times w$, where *r* is the number of rows and *w* is the number of columns in the rectangular grid as shown in Fig. 2. Since the signaling traffic for location update depends on the size of write sets, and the signaling traffic for location query depends on the size of read sets, it is desirable to make both the write sets and read sets as small as possible. However, for a given number of databases in the system, the parameters *r* and *w* cannot be adjusted independently. If one is increased, the other must be decreased and vice versa, so we can not make both of them as small as desired. By changing *r* and *w*, multiple collections of variously sized write sets and read sets can be constructed. Each collection corresponds to a pair of parameters, *w* and *r*. We can use different sizes of write sets and read sets to balance the signaling cost for location update and query. For the case of $N = 1$, only one write set and read set can be constructed, either includes one database. This case has the lowest signaling overhead for location update and query, since write set and read set contains only one database. However, this case is unrealistic in a large system. Since location update and query for all mobile users in the system have to access this single database, this may overload this database and cause much delay. Another reason is that the system is very vulnerable to the failure of this database. For a real system, the designers must take all these issues into consideration and place a certain number of databases in the system.

Based on the above discussion, we propose a *rectangular grid-based scheme* to construct write sets and read sets. We assume that the total number of databases in the system, N, is an even power of 2. That is, $N = 2^{2k}$ where k is an integer. Several $r \times w$ rectangular grids with different length and width can be constructed, while keeping $N = r \times w$. For example, we want to construct three collections of write sets and read sets for $N = 16$. These 16 databases are logically organized into three rectangular grids as shown in Fig. 1 through Fig. 3. Each grid point represents a database. For each figure, we let the databases on each row constitute a write set, and the databases on each column constitute a read set. Thus, for each figure, each write set and each read set has one database in common, and any database belongs to one write set and one read set. In this way, a collection of write sets of size 4 and read sets of size 4 is formed by Fig. 1. A collection of write sets of size 8 and read sets of size 2 is constructed by Fig. 2. A collection of write sets of size 2 and read sets of size 8 is formed by Fig. 3. These three collections of write sets and read sets are logically arranged into three steps labeled step 1 through step 3 such that the sizes of read sets at different steps are in ascending order. That is, the collection of write sets of size 8 and read sets of size 2 formed by Fig. 2 corresponds to step 1. The collection of write sets of 4 and read sets of 4 formed by Fig. 1 corresponds to step 2. The collection of write sets of size 2 and read sets of size 8 formed by Fig. 3 corresponds to step 3. Let w_i and r_i represent the sizes of write sets and read sets at step i respectively, then we have,

$$r_i = 2 \cdot r_{i-1} \qquad i=2, 3. \qquad (1)$$

$$w_i = \frac{1}{2} \cdot w_{i-1} \qquad i=2, 3. \qquad (2)$$

Furthermore, we can see that Fig. 2 can be realized by moving the bottom 2 rows of Fig. 1 to the right of the top 2 rows of Fig. 1. While, the Fig. 3 can be obtained by moving the right two columns of Fig. 1 below the left two columns of Fig. 1. Thus, for any given database A shown in these three figures, the read set (at a lower step) to which database A belongs is a subset of the read set (at a higher step) to which database A belongs. The write set (at a higher step) to which database A belongs is a subset of the write set at a lower step to which database A belongs. This feature can be used to further reduce the average number of databases queried to find a mobile user's current LA. The properties of the write sets and read sets constructed using this method are summarized below:

- Each step includes one or more write sets (or read sets), the write sets (or read sets) have the same size at the same step.
- Each database belongs to one write set and one read set at each step.
- Any write set and any read set at the same step intersect.
- The size of read sets at step i is twice the size of read sets at steps i-1, and the size of write sets at step i is half of the size of write sets at step i-1.
- For any given database, the read set (at a lower step) to which the given database belongs is a subset of the read set (at a higher step) to which the given database belongs, and the write set (at a higher step) to which the given database belongs is a subset of the write set (at a lower step) to which the given database belongs.

For $N = 2^{2k}$ and k is an integer, at most $2k+1$ steps of different sizes of write sets and read sets can be constructed using above-described method. If N is not an even power of 2, some of the databases can be used twice in the rectangular grids to make the number of total grid points be an even power of 2. So, the proposed construction method can be still used.

3. System Description of the Proposed Scheme

We assume that the entire service area is divided into many cells, each served by a base station. Mobile users in a cell communicate with the base station in the cell. The cells are in turn grouped into LAs such that each LA consists of a cluster cells. The base stations of all cells in a group of LAs are connected to a MSC that provides switching functions and coordinates location registration and call delivery. Each MSC is associated with a database, which is used to store mobile user location information. Each database is assigned a unique ID. Since each MSC has one database, a database can also be identified by the MSC with which it is associated. The MSCs in the system are connected to the backbone network. Mobile users are free to move in the service area. Each time a mobile user enters a new LA, a location update is required. When a call arrives, the system queries appropriate databases to get the current LA of the called mobile user. Paging in that LA is then performed. Many LA design schemes and paging strategies have been proposed [11], [12], [13], [14]. In this paper, we assumed that the LA design scheme and paging strategy have already been decided.

In the proposed scheme, we assume there are a total of N databases in the system, and these N databases are logically organized into n collections of write sets and read sets using the rectangular grid-based construction method discussed in section 2. The n collections of write sets and read sets are in turn logically arranged into n steps, with each collection corresponding to one step. Each database in the system belongs to one write set and one read set at each step simultaneously. The write sets and read sets at different steps differ in size. Any write set and any read set at the same step intersect. We also assume that the steps are labeled 1 through n such that read sets at higher steps are larger than read sets at lower steps. For any given database, the read set (at a lower step) to which the given database belongs is a subset of the read set (at a higher step) to which the given database belongs. The write set (at a higher step) to which the given database belongs is a subset of the write set (at a lower step) to which the given database belongs. Let w_i and r_i denote the sizes of write sets and read sets at step i respectively, then we have $r_1 < r_2 < \cdots < r_n$. Since smaller read sets imply larger write sets, we have $w_1 > w_2 > \cdots > w_n$.

The system's n steps of write sets and read sets can be implemented by having each MSC maintain n *write lists* and n *read lists*. Entries of a write list are the IDs of the databases in a write set of a certain step to which the database associated to this MSC belongs. Entries of a read list are the IDs of the databases in a read set of a certain step to which the database associated to this MSC belongs. For any given database, the write set (at step i) to which this given database belongs is just the union of databases whose IDs are included in the i-th write list of its associated MSC. The read set (at step i) to which this given database belongs is just the union of databases whose IDs are included in the i-th read list of its associated MSC. This write (or read) set is called *write* (or *read*) *set* of the MSC at step i. With this implementation scheme, a MSC can easily identifies which databases are in its write set or read set at a certain step.

4. Location Update and Location Query Mechanisms for the Proposed Scheme

In the proposed scheme, mobile users with different mobility and call activities will register at different steps. A mobile user is said to *register* at step i when its current LA is stored in a write set at step i. Step i is called the mobile user's *registration step.* In order to classify mobile users, we characterize the classes of mobile users by their *call-to-mobility ratio* (CMR) in a manner similar to that in [3], [4].

Let T_D denote the (random) time that a mobile user spends in a LA. This is called the *dwell time* in a LA. A large T_D implies a slow mobility for the mobile user. Then a mobile user's mobility is characterized by V is defined as

$$V = \frac{1}{T_D} \tag{3}$$

which is just the number of location updates per unit time.

If a mobile user receives calls at an average rate λ, then the CMR of the mobile user, denoted by ρ, is

$$\rho = \frac{\lambda}{V} = \lambda \cdot T_D \tag{4}$$

From (4), we see that the CMR of a mobile user can be also viewed as the number of calls received during the time period that the mobile user remains in a LA. The user itself can estimate its CMR. In the proposed scheme, the CMR of a mobile user is used to determine the registration step of the mobile user. This will be discussed subsequently in more details.

4.1. Location Update Procedure

We assume that each base station in the service area periodically broadcasts its base station ID through its broadcast channel. The base station ID is the same as the ID of the cell in which the base station is located. A mobile user monitors the strongest broadcast channel as it moves. Every time a mobile user enters a cell that belongs to a new LA, a location update is initiated. When updating its LA, the mobile user sends a location update request to the base station of the cell that it is entering. The base station forwards this request to the MSC to which it is connected. Since each MSC has only one write set and one read set at a given step, the write set for this location update is solely determined by the mobile user's CMR and the base station ID of the cell in which the location update is initiated. This location update procedure is summarized as follows:

1. The mobile user sends an update request to the base station of the cell in which the location update is initiated. This request includes the following information: the user's ID; the ID of the new LA; the user's current CMR; the user's previous CMR when the user last updated its LA; and the ID of the base station in which the user last initiated a location update, Previous_BS_ID. This requires that each mobile user store its previous CMR and Previous_BS_ID in its local memory.
2. The base station forwards this request to the MSC to which it is connected. This MSC is called the current host MSC of the mobile user. The current host MSC checks if the base station with Previous_BS_ID is one that is connected with the MSC itself. If the base station with Previous_BS_ID is, the current host MSC will determine the previous registration step of the mobile user according to the user's previous CMR. The current host MSC then sends a DELETE message to its write set at the previous registration step to delete the entries (in all databases of the write set) that contain the user's previous LA. If the base station with Previous_BS_ID is not connected to the current host MSC, the current host MSC will send the user's ID and previous CMR to the previous MSC. The previous MSC will determine the write set in which the user's previous LA is stored and delete the previous location information of the mobile user.
3. The current host MSC determines the new registration step of the mobile user according to the user's current CMR. Then, the current host MSC sends an ADD message to the databases in its write set at the new registration step. On receiving an ADD message, each database adds a new entry to itself that contains the user's ID and the ID of the user's new LA.

4.2. Location Query Procedure

When an incoming call arrives, the current LA of the called mobile user can be obtained from a read set at a certain step. The following operations are needed to get the current LA of the called mobile user.

1. The caller, if it is a mobile, sends a call setup request to the base station of the cell in which the caller is located. This request includes the called mobile user's ID. The base station forwards this call setup request to the MSC to which it is connected. If the caller is a fixed wireline user, the wireline network will forward the call setup request to the nearby MSC. In either case, this MSC is called the current host MSC of the caller.
2. The current host MSC sends a QUERY message that contains the called mobile user's ID to the databases in its read set at step 1. If any database in the read set contains the ID of the called mobile user's current LA, it sends this LA ID to the host MSC in its RESPONSE message. Otherwise, the databases send a NULL message in RESPONSE.
3. On receiving RESPONSE messages, the current host MSC checks it. If the ID of the called mobile user's LA returned, the query procedure stops. Otherwise, the host MSC sends a QUERY message to the databases in its read set at the next higher step. Those databases that also belong to read sets at the lower levels and have already been queried need not be queried again. This procedure continues until the ID of the called mobile user's current LA is obtained.

The location query procedure described above ensures that the system can get the ID of the called mobile user's current LA after the completion of the procedure, since the query procedure eventually goes to the step at which the called mobile user is registered. Since any write set and any read set at the same step intersect, the ID of the called mobile user's current LA can be obtained in any read set at the called user's registered step. After the called mobile user's current LA is obtained, the system will page that user in the user's current LA to get the user's *cell* location, then a call connection can be set up through the base station in that *cell*.

5. Determination of Registration Steps

As described earlier, both location update and location query generate a certain amount of signaling traffic, these signaling traffic consumes some wireline transmission bandwidth on the backbone network. As a result, there are some costs associated with both location update and location query. For the proposed scheme, it is clear that if a mobile user is registered at step 1, the system needs to update the user's LA in a large write set. The cost for location update is high. However, the cost for location query will be low, since a location query only needs to query a read set of step 1 that is small to get the ID of the user's current LA. If a mobile user is registered at the highest step, the system needs to update the user's LA in a relative small write set. In this case, the cost for location update is small. However, the cost for location query in this case will be high, since the system may need to go to the highest step and query a read set which is large to obtain the ID of the user's current LA. Therefore, a tradeoff exists between the location update cost and the location query cost in the proposed scheme. It is desirable to select an optimal registration step for each mobile user such that the *combined* signaling cost of location update and query is minimized.

The system has n steps of write sets and read sets, each of which is characterized by the sizes of write sets and read sets, w_i and r_i, i=1, 2, 3, . . . n. The rates of location update and query for a mobile user are related to the dwell time in a LA, T_D, and call arrival rate, λ. So, each mobile user should choose the registration step to update its LA according to these two parameters. A mobile user travels in the service area with changing mobility and changing call arrival rate, so the registration step for the mobile user should be dynamically changed.

In order to find the optimal registration step for a given mobile user and analyze the performance, the following parameters are assumed:

- N: the total number of databases in the system.
- n: the total number of steps of write sets and read sets in the proposed scheme.
- w_i : the number of databases contained in a write set at step i.
- r_i : the number of databases contained in a read set at step i.
- λ: the average call arrival rate to a given mobile user.

- T_D : the dwell time of a given mobile user in a LA.
- ρ: the CMR of a given mobile user.
- $p_{i,j}$: the probability with which a mobile user's current LA can be found in a read set at step j when the mobile user is registered at step i.
- U : the average signaling cost for updating a LA ID in one database. This cost is a function of the amount of message transmitted and the distance traveled by the transmitted messages.
- D : the average signaling cost for deleting a LA ID in one database. This cost is a function of the amount of message transmitted and the distance traveled by the transmitted messages.
- Q : the average signaling cost for querying one database. This cost is a function of the amount of message transmitted and the distance traveled by the transmitted messages.
- C_t : the *combined* cost for location update and location query for a given mobile user.

From the write sets and read sets construction, we have,

$$N = r_i \times w_i \qquad i=1, 2, 3, \dots n \tag{5}$$

$$r_i = 2 \cdot r_{i-1} \qquad i=2, 3, 4, \dots n \tag{6}$$

$$w_i = \frac{1}{2} \cdot w_{i-1} \qquad i=2, 3, 4, \dots n \tag{7}$$

Suppose a mobile user with parameters T_D and λ is registered at step i, i=1, 2, 3, ...n. Then, the ID of the mobile user's current LA is stored in a write set at step i. Thus, the current LA of the mobile user can be found in a read set at step i with probability 1. Recall that a read set at a lower step is a subset of a read set at higher steps, and the size of read sets at step i is twice the size of read sets at step i-1. Therefore,

$$p_{i,j} = 1 \qquad j= i, i+1, i+2, \dots n \tag{8}$$

$$p_{i,j} = \frac{1}{2^{i-j}} \qquad j= 1, 2, 3, \dots i-1 \tag{9}$$

For each location update, the system will update the user's LA in a write set at step *i*, and deletes the ID of the user's previous LA in a write set at step *i*. Since a write set at step *i* includes w_i databases and the mobile user performs $(1/T_D)$ location updates per unit time, the signaling cost for location update per unit time per user, C_u, is

$$C_u = (U \cdot w_i + D \cdot w_i) \cdot \frac{1}{T_D} = (U + D) \cdot \frac{w_i}{T_D} = (U + D) \cdot \frac{w_1}{2^{i-1} \cdot T_D} \tag{10}$$

When there is an incoming call to this mobile user, the system performs a location query as described in section 4.2. According to the query procedure, we can find the average number of databases queried, $\overline{M}$, to find the called mobile user's LA is

$$\overline{M} = r_1 + (r_2 - r_1) \cdot (1 - p_{i,1}) + (r_3 - r_2) \cdot (1 - P_{i,2}) + \ldots + (r_i - r_{i-1}) \cdot (1 - p_{i,i-1}) \tag{11}$$

Substitute $p_{i,j}$ into equation (11), we have

$$\begin{aligned}\overline{M} &= r_1 + \sum_{j=2}^{i} (r_j - r_{j-1}) \cdot (1 - p_{i,j-1}) = r_1 + \sum_{j=2}^{i} (2^{j-1} \cdot r_1 - 2^{j-2} \cdot r_1) \cdot (1 - \frac{1}{2^{i-j+1}}) \\ &= r_1 + r_1 \cdot \sum_{j=2}^{i} 2^{j-2} \cdot (1 - \frac{1}{2^{i-j+1}})\end{aligned} \tag{12}$$

So, the average signaling cost for location query of the mobile user per unit time, C_q, is

$$C_q = \lambda \cdot \overline{M} \cdot Q = \lambda \cdot Q \cdot r_1 \cdot [1 + \sum_{j=2}^{i} 2^{j-2} \cdot (1 - \frac{1}{2^{i-j+1}})] \tag{13}$$

Therefore, the *combined* cost for location update and query per user per unit time, C_t, is

$$C_t = C_u + C_q \tag{14}$$

That is,

$$\begin{aligned}C_t &= (U + D) \cdot \frac{w_1}{2^{i-1} \cdot T_D} + \lambda \cdot Q \cdot r_1 \cdot [1 + \sum_{j=2}^{i} 2^{j-2} \cdot (1 - \frac{1}{2^{i-j+1}})] \\ &= \lambda \cdot Q \cdot \left\{ (\frac{U}{Q} + \frac{D}{Q}) \cdot \frac{N}{2^{i-1} \cdot r_1 \cdot \rho} + r_1 \cdot [1 + \sum_{j=2}^{i} 2^{j-2} \cdot (1 - \frac{1}{2^{i-j+1}})] \right\}\end{aligned} \tag{15}$$

The objective here is to select step i such that the *combined* cost for location update and query, C_i, is minimum for a given ρ. The step that minimizes the combined signaling cost, C_i, is called the *optimal registration step* of the mobile user. From equation (15), we see that the optimal registration step for a mobile user depends on the user's CMR, so the optimal registration step for a given mobile user with CMR of ρ can be numerically solved by minimizing C_i with respect to i.

In order to compute the *combined* cost for location update and query, we have first to determine U, D and Q in equation (15). According to the proposed location update and location query mechanisms, the UPDATE message and DELETE message are both one-way messages, while QUERY message is round-trip. It is reasonable to assume that $\frac{U}{Q} = \frac{D}{Q} = \frac{1}{2}$ for numerical purpose. We consider a system that has a total of $N = 2^{10} = 1024$ databases in the service area. These N databases are logically organized into total n steps of write sets and read sets according to the write sets and read sets construction method described in section 2. From equations (6) and (7), we know that the size of read sets at step i is twice the size of read sets at steps i-1, and the size of write sets at step i is half of the size of write sets at step i-1. So, for a given n, once the sizes of write sets and read sets at step 1 (i.e., w_1 and r_1) are given, the sizes of write sets and read sets at other steps are determined. Fig. 4 (a)-(d) show the optimal registration steps of a mobile user (that is, the best CMR break points) for the cases of n=2, 3, 4 and 5, respectively. For the cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$ are chosen. For the cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$ are assumed. We can see that a mobile user with high CMR registers at a lower step, i.e., a step with large-size write sets and small-size read sets. While, a mobile user with low CMR registers at a higher step, i.e., a step with small-size write sets and large-size read sets.

6. Performance Analysis

6.1. Combined Cost of Location Update and Query

Recall that for $N = 2^{2k}$ and k is an integer, at most $(2 \cdot k + 1)$ steps with different sizes of write sets and read sets can be constructed using the proposed write sets and read sets construction method. If the number of steps, n, is smaller than $(2 \cdot k + 1)$, there are many options to select n steps from the total of $(2 \cdot k + 1)$ steps. The system performance greatly depends on the selection of these n steps. The best scenario is to classify all mobile users into n categories, and then select these n steps according to the CMRs of these n classes of mobile users such that the overall signaling cost for location update and query is minimized. This needs some actual statistics from the industries. For numerical purpose, we here select these n steps such that the sizes of write sets and read sets at these steps are around $\sqrt{N} = 2^k$. In this way, the sum of the number of databases in a write set and the number of the databases in a read set at each step are smaller than that of any other selections. Therefore, the total number of databases accessed for location update and query per unit time may be reduced. We assume that both the dwell time of a mobile user in a LA, T_D, and the call arrival rate of the mobile user, λ, are negative exponentially distributed (n.e.d.) random variables with mean value of $\overline{T}_D$ and $\overline{\lambda}$ respectively. The value of $\overline{T}_D$ can be considered as the average dwell time of a mobile user in all LAs, while the value, T_D, can be thought as the dwell time of a mobile user in a single LA. Similarly, the value of $\overline{\lambda}$ can be treated as the average call arrival rate over all mobile users or a group of mobile users, while the value of λ can be considered as the call arrival rate for a given mobile user. Therefore, their probability density functions (pdf) are as follows.

$$f(T_D) = \frac{1}{\overline{T}_D} \cdot e^{-T_D/\overline{T}_D} \qquad T_D \in [\,0, \infty) \qquad (16)$$

$$f(\lambda) = \frac{1}{\overline{\lambda}} \cdot e^{-\lambda/\overline{\lambda}} \qquad \lambda \in [\,0, \infty) \qquad (17)$$

Note that, in [16], we assumed λ is a fixed parameter for a given mobile user, and that the CMR of the given mobile user, ρ, is a negative exponentially distributed random variable with mean $\overline{\rho} = \lambda \cdot \overline{T}_D$.

For a n-step of write sets and read sets system with N databases, the optimal registration step for a mobile user with CMR of ρ can be determined according to the discussion in last section. We assume that the optimal registration step for a given mobile user is step $i(\rho)$, which is ρ dependent. According to equation (15), the combined signaling cost of location update and query for the mobile user with CMR, ρ, per unit time is given by

$$C_{i(\rho)} = \lambda \cdot Q \cdot \left\{ (\frac{U}{Q} + \frac{D}{Q}) \cdot \frac{N}{2^{i(\rho)-1} \cdot r_1 \cdot \rho} + r_1 \cdot [1 + \sum_{j=2}^{i(\rho)} 2^{j-2} \cdot (1 - \frac{1}{2^{i(\rho)-j+1}})] \right\} \quad (18)$$

The combined cost, $C_{i(\rho)}$, is a function of λ and T_D, because $\rho = \lambda \cdot T_D$. Thus, the average combined signaling cost of location update and query for the mobile user per unit time in the n-step scheme, C_{Ave}, is found by averaging over λ and T_D. That is,

$$C_{Ave} = \int_0^\infty \int_0^\infty C_{i(\rho)} \cdot f(\lambda) \cdot f(T_D) \cdot d\lambda \cdot dT_D \quad (19)$$

Since we want to compare the proposed scheme with the scheme having the same number of databases that are logically arranged as a single collection of sets (quorums) every two of which intersect. The quorums are assumed to be constructed using the square grid method. We call the latter scheme the quorum-based scheme. We need to find the combined signaling cost of the location update and query for a mobile user per unit time in the quorum-based scheme. The quorum-based scheme can be considered as a special case of our proposed scheme, i.e, the case in which n=1, and both the size of write sets and the size of read sets are $(2 \cdot \sqrt{N} - 1)$. So the performance of the quorum-based scheme can be found as follows.

The combined signaling cost of location update and query for a mobile user with CMR, ρ, per unit time in the quorum-based scheme, C', is,

$$C' = \lambda \cdot Q \cdot (2 \cdot \sqrt{N} - 1) \cdot [(\frac{U}{Q} + \frac{D}{Q}) \cdot \frac{1}{\rho} + 1] \quad (20)$$

The combined cost, C', is also a function of λ and T_D. Therefore, the average combined signaling cost of location update and query for the mobile user per unit time, C'_{Ave}, is given by

$$C'_{Ave} = \int_0^\infty \int_0^\infty C' \cdot f(\lambda) \cdot f(T_D) \cdot d\lambda \cdot dT_D \quad (21)$$

We are interested in the ratio of the average combined signaling costs between the proposed scheme and the quorum-based scheme, C_{Ave} / C'_{Ave}, which reflects the savings in the combined cost by using the proposed scheme. Let R denote the ratio, then,

$$R = \frac{C_{Ave}}{C'_{Ave}} \quad (22)$$

The ratio, R, can be numerically computed for the given parameters and conditions. We consider an n-step write sets and read sets scheme with N=1024. The numerical results are shown in Fig. 5 and Fig.6. In Fig. 5, the ratio, R, is plotted as a function of $\overline{T}_D$ for given $\overline{\lambda}$. The ratio, R, is plotted as a function of $\overline{\lambda}$ for given $\overline{T}_D$ in Fig. 6. These figures show the results for the cases of n=1, 2, 3, 4 and 5, where n is the total number of registration steps. For the case of n=1, $w_1 = 32$ and $r_1 = 32$ are chosen. For the cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$ are chosen. For the cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$ are assumed. From the figures, we see that the proposed scheme can result in almost 50% savings in average combined cost over the quorum-based scheme when n=1. The average combined signaling cost for the proposed scheme can be further reduced by increasing the number of steps, n.

6.2. Time for Location Update and Query

In the procedure of location update, two multicast messages, DELETE message and ADD message, are needed to update a mobile user's LA in a write set, each of which is a one-way message and can be simultaneously sent to all databases in a write set. This location update procedure is similar to that of quorum-based location management scheme. The location update can be completed in one round-trip message delay. This is comparable to the quorum-based location management scheme and centralized HLR-VLR scheme.

For each location query, the system sequentially queries a read set of each step starting from step 1 until the called mobile user's current LA is obtained. From the proposed location query procedure described in section 4.2, it is clear that a query to one read set can be completed in one round-trip message delay. We assume that one round-trip message delay is t_r. Recall that when a mobile user is registered at step *i*, the probability with which a mobile user's current LA can be found in a read set at step *j* is $p_{i,j}$, which can be obtained form equations (8) and (9). Therefore, when a mobile user is registered at step *i*, the time to obtain a mobile user's current LA, $\bar{t}_i$, is given by

$$\begin{aligned} \bar{t}_i &= t_r + t_r \cdot (1 - p_{i,1}) + t_r \cdot (1 - p_{i,2}) + \ldots + t_r \cdot (1 - p_{i,i-1}) \\ &= t_r \cdot \left[1 + \sum_{j=1}^{i-1} (1 - p_{i,j}) \right] \end{aligned} \tag{23}$$

We assume the probability that a mobile user is registered at step *i* is P_i. For an *n*-step of write sets and read sets scheme, the average time to obtain a mobile user LA, $\bar{T}_q$, is,

$$\bar{T}_q = \sum_{i=1}^{n} P_i \cdot \bar{t}_i = \sum_{i=1}^{n} \left\{ P_i \cdot t_r \cdot \left[1 + \sum_{j=1}^{i-1} (1 - p_{i,j}) \right] \right\} \tag{24}$$

$\bar{T}_q$ is called the *average query delay* of a mobile user.

We assume that ρ_{i-1} and ρ_i are respectively the lower and upper bound of the interval of CMR, ρ, in which a mobile user is registered at step *i*. For a mobile user having CMR, ρ, P_i can be found as follows,

$$P_i = \frac{\int_{\rho_{i-1}}^{\rho_i} f(\rho) \cdot d\rho}{\int_0^{\infty} f(\rho) \cdot d\rho} = \frac{\int_0^{\rho_i} f(\rho) \cdot d\rho - \int_0^{\rho_{i-1}} f(\rho) \cdot d\rho}{\int_0^{\infty} f(\rho) \cdot d\rho} \tag{25}$$

Where, $f(\rho)$ is the pdf of ρ.

Since $\rho = \lambda \cdot T_D$, we have,

$$P_i = \frac{\int_0^\infty \int_0^{\rho_i/\lambda} f(\lambda)\cdot f(T_D)\cdot d\lambda \cdot dT_D - \int_0^\infty \int_0^{\rho_{i-1}/\lambda} f(\lambda)\cdot f(T_D)\cdot d\lambda \cdot dT_D}{\int_0^\infty \int_0^\infty f(\lambda)\cdot f(T_D)\cdot d\lambda \cdot dT_D} \tag{26}$$

Since λ and T_D are independent to each other, and $\int_0^\infty f(\lambda)\cdot d\lambda = \int_0^\infty f(T_D)\cdot dT_D = 1$, we obtain,

$$P_i = \int_0^\infty \int_0^{\rho_i/\lambda} f(\lambda)\cdot f(T_D)\cdot d\lambda \cdot dT_D - \int_0^\infty \int_0^{\rho_{i-1}/\lambda} f(\lambda)\cdot f(T_D)\cdot d\lambda \cdot dT_D \tag{27}$$

The average query delay, $\overline{T_q}$, can be numerically computed for the given parameters and conditions. We consider an *n*-step write sets and read sets scheme with *N*=1024. The numerical results are shown in Fig. 7 and Fig. 8. In Fig. 7, the average query delay, $\overline{T_q}$, is plotted as a function of $\overline{T_D}$ for given $\overline{\lambda}$. The average query delay, $\overline{T_q}$, is plotted as a function of $\overline{\lambda}$ for given $\overline{T_D}$ in Fig. 8. These figures show the results for the cases of *n*=1, 2, 3, 4 and 5. For the case of *n*=1, $w_1 = 32$ and $r_1 = 32$ are chosen. For the cases of *n*=2 and 3, $w_1 = 64$ and $r_1 = 16$ are chosen. For the cases of *n*=4 and 5, $w_1 = 128$ and $r_1 = 8$ are assumed. From the figures, we see that the average query delay of the proposed scheme increases with the total number of steps, *n*, increasing. This is because that the average number of query steps increases as *n* increases. When compared with the quorum-based scheme, the proposed scheme has a higher average query delay. The query delay is one round-trip message delay in the quorum-based scheme, since the system only needs to query one quorum to get a mobile user's current LA. In the HLR-VLR scheme, the system needs to first query a mobile user's HLR, then the user's serving VLR to get the mobile user's current LA. The query in HLR-VLR scheme has a two round-trip message delay. So, when compared with the HLR-VLR scheme, the proposed scheme has a smaller query delay under some conditions, a higher query delay under the other conditions. For example, from Fig. 7(b) we can see that for the case of the total number of registration steps in the proposed scheme, *n*, is equal to 4, the proposed scheme has a average query delay smaller than 2 when $\overline{T}_D$ is greater than 3 hours. Otherwise, the average query delay of the proposed scheme is greater than 2.

To reduce the average query delay of the proposed scheme, one possible method is that the system start querying a read set at a step higher than step 1. This can increase the probability of finding a mobile user's current LA at each query. However, this may increase the average query signaling cost, since the system queries a relative larger read set at each query step. So, in reality, the tradeoff between the average query delay and average query signaling cost should be considered.

7. Conclusions

A scheme in which the databases are logically organized into multi-step write sets and read sets for location management in personal communication systems was proposed and analyzed. The implementation was also discussed. Based on a mobile user's CMR, the mobile user is registered at an optimal step such that the combined signaling cost of location update and query on the backbone network for the mobile user is minimized. Depending on the write sets and read sets construction methods, the system can result in substantial reduction in the combined signaling cost for location update and query when compared with a scheme having the same number of databases that are logically arranged as a single collection of sets every two of which intersect.

References

[1] S. Mohan and R. Jain, "Two user location strategies for personal communications services," IEEE *personal Communications*, vol. 1. No. 1, 1994, pp.42-50.

[2] I. F. Akyildiz and J. S. M. Ho, "On location management for personal communications networks," IEEE *Communication Magazine*, Sept.1996, pp. 138-145.

[3] R. Jain and Y.-B. Lin, "An auxiliary user location strategy employing forwarding pointers to reduce network impact of PCS," *ACM-Baltzer J. Wireless Networks*, vol. 1, no. 2, 1995, pp. 197-210.

[4] R. Jain, Y.-B. Lin and S. Mohan, "A caching strategy to reduce network impacts of PCS," IEEE *JASC*, vol. 12, no. 8, Oct., 1994, pp. 1434-1444.

[5] R. Prakash, Z. Haas and M. Singhal, " Load-balanced location management for mobile systems using dynamic hashing and quorums," submitted for publication.

[6] Z. Haas and B. Liang, "Ad-hoc location management using quorum systems," *IEEE/ACM Trans. Networking*, vol. 7, no. 2, Apr. 1999. pp. 228-240.

[7] B. Awerbuch and D. Peleg, " Online tracking of mobile users," *Journal of the ACM*, vol. 42, no. 5, Sept. 1995, pp.1021-1058.

[8] M. Naor and A. Wool, " The load, capacity, and availability of quorum systems," *SIAM J. Comput.*, vol. 27, no. 2, April 1998. pp. 423-447.

[9] D. Peleg and A. Wool, "Crumbling walls: a class of practical and efficient quorum systems," *Proc. of the 14th ACM Symposium on Principles of Distributed Computing*, Aug. 1995. pp. 120-129.

[10] D. Peleg and A. Wool, "How to be an efficient snoop, or the probe complexity of quorum systems, " *Proc. of the 15th ACM Symposium on Principles of Distributed Computing*, May 1996. pp. 290-299.

[11] L.-R. Hu and S. S. Rappaport, "An adaptive location management scheme for global personal communications," IEE *Proc. Commun.*, vol. 144, no. 1, Feb. 1997. pp. 54-60.

[12] D. Gu and S.S. Rappaport, "A dynamic location tracking strategy for mobile communication systems," *Proc. VTC '98*. Ottawa, Canada. May, 1998, pp.259-263.

[13] D. Gu and S.S. Rappaport, "Mobile user registration in cellular systems with overlapping location areas," *Proc. VTC'99*. Houston, USA. May 1999, pp. 1.5, 1, 1-5.

[14] C. Rose and R. Yates, "Minimizing the average cost of paging underlay constraints," *ACM-Baltzer J. Wireless Networks*, vol. 1, no. 2, 1995, pp. 211-219.

[15] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. VT-35, no. 3, Aug. 1986, pp. 77-92.

[16] D. Gu, *Location Management for Mobile Cellular Communications*, Ph.D. thesis, Aug. 1999.

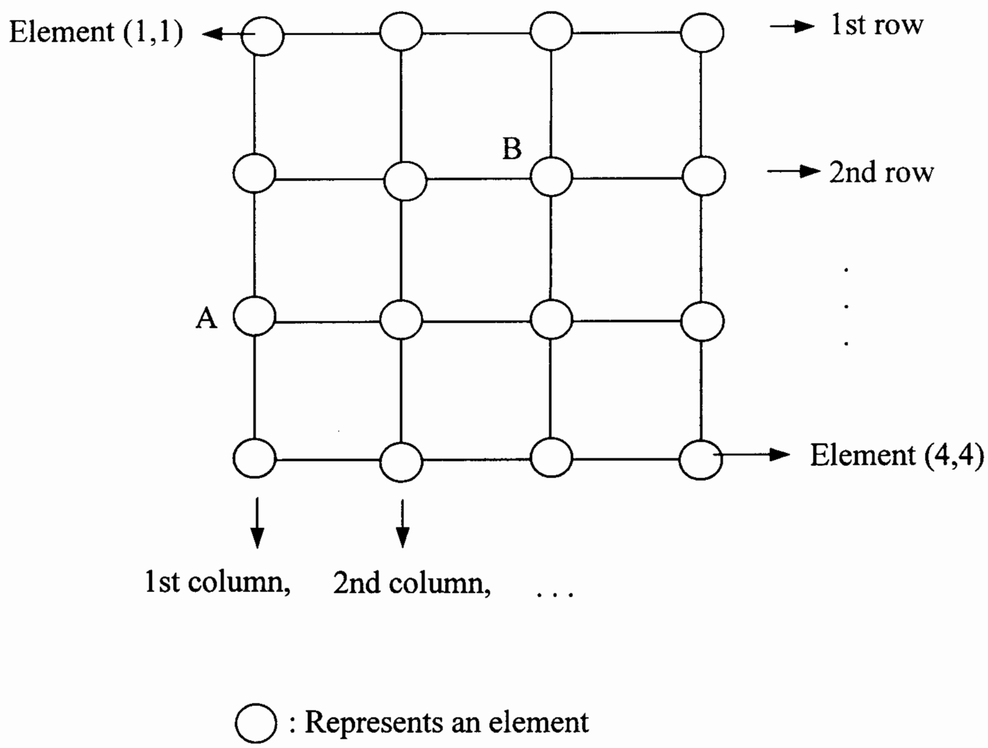


Fig. 1: A square grid-based quorum construction scheme with N=16, r=w=t=4.

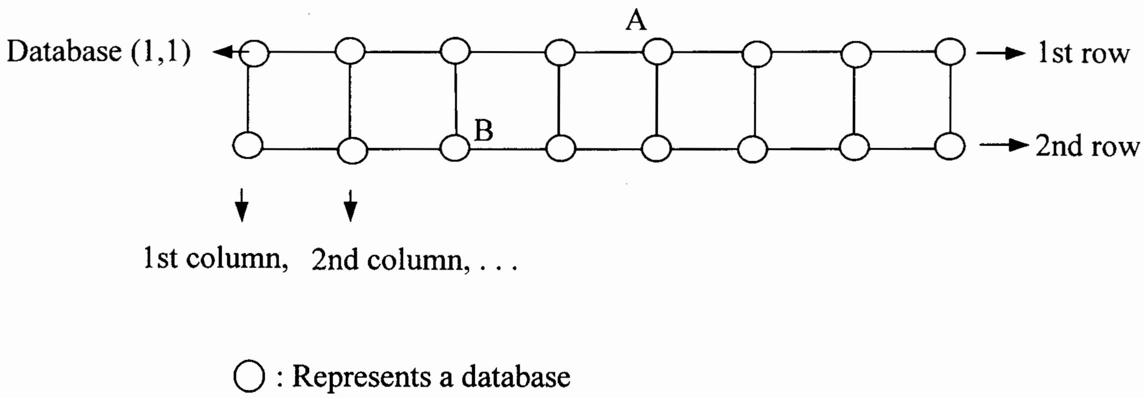


Fig. 2: A rectangular grid-based construction scheme for write sets and read sets with N=16, r=2, w=8.

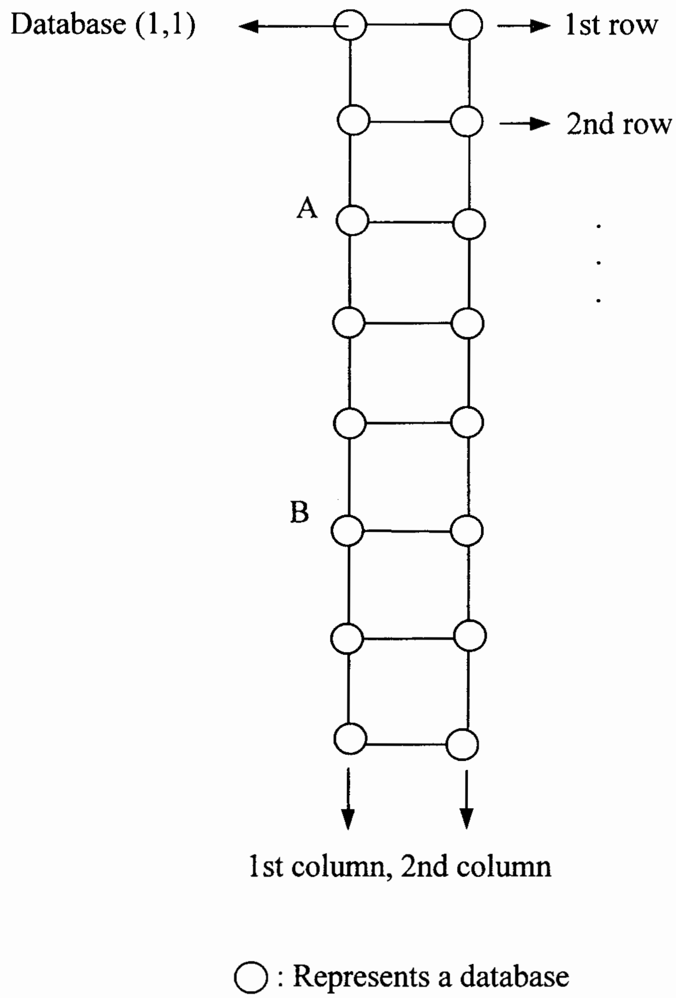
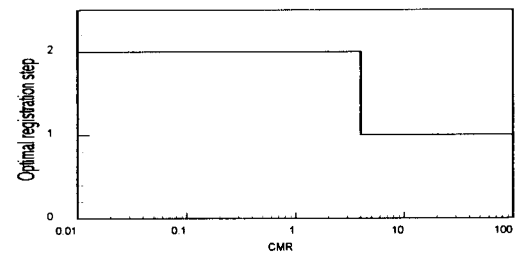
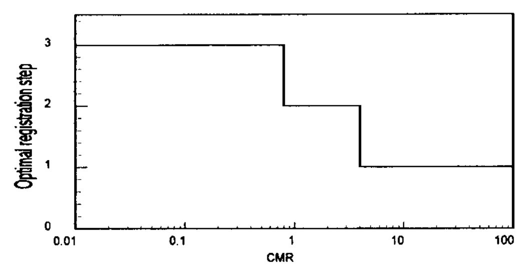


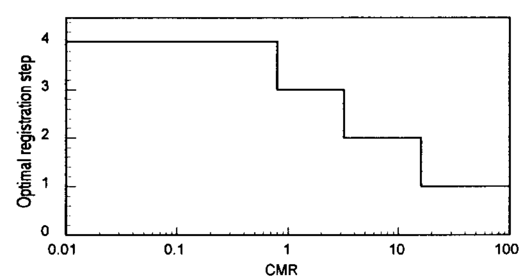
Fig. 3: A rectangular grid-based construction scheme for write sets and read sets with N=16, r=8, w=2.



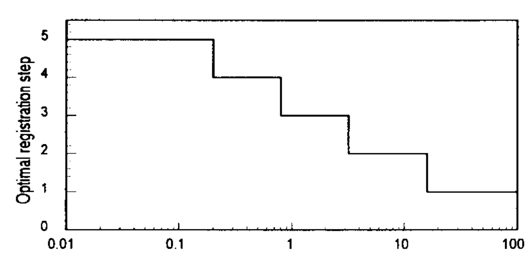
(a) The number of steps, n=2



(b) The number of steps, n=3

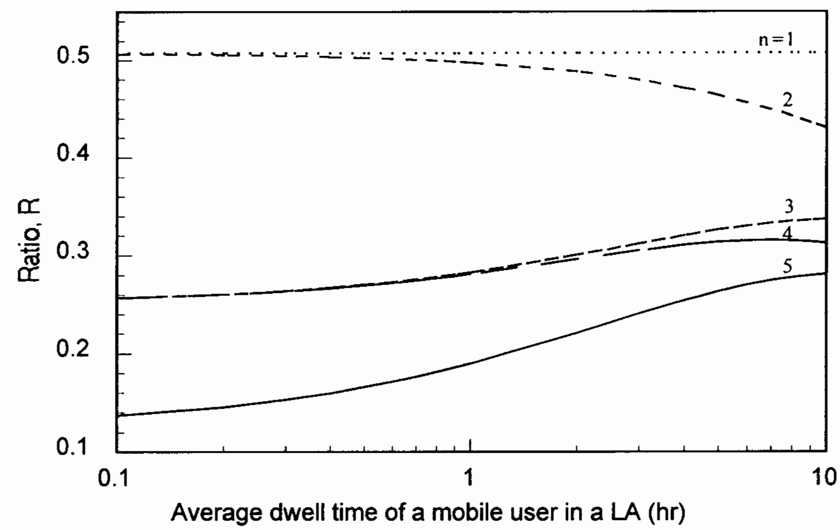


(c) The number of steps, n=4

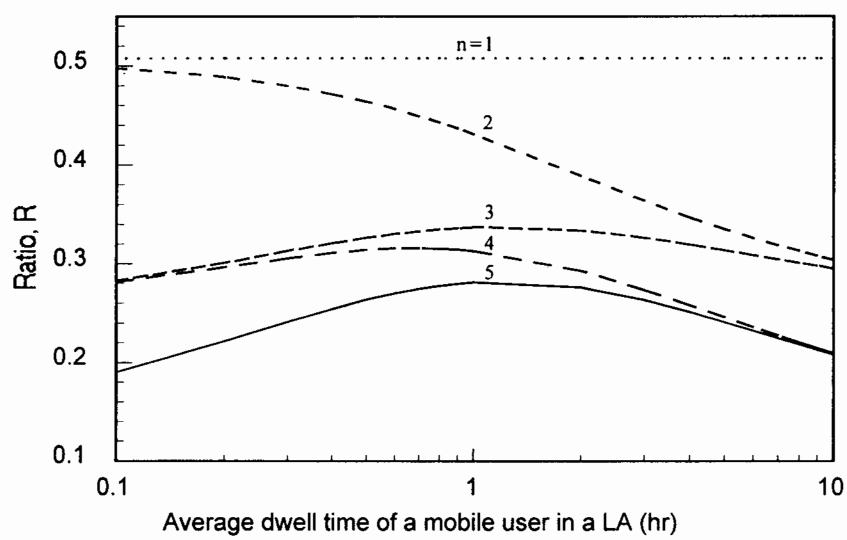


(d) The number of steps, n=5

Fig. 4: The optimal registration steps for a mobile user.
Parameters: N=1024.
For cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$.
For cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$.

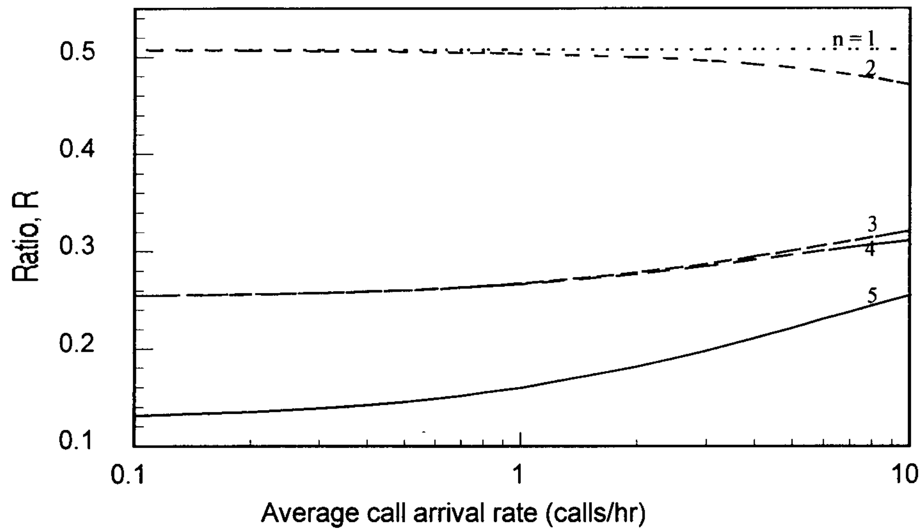


(a) Average call arrival rate, $\overline{\lambda} = 0.5$ calls/hr

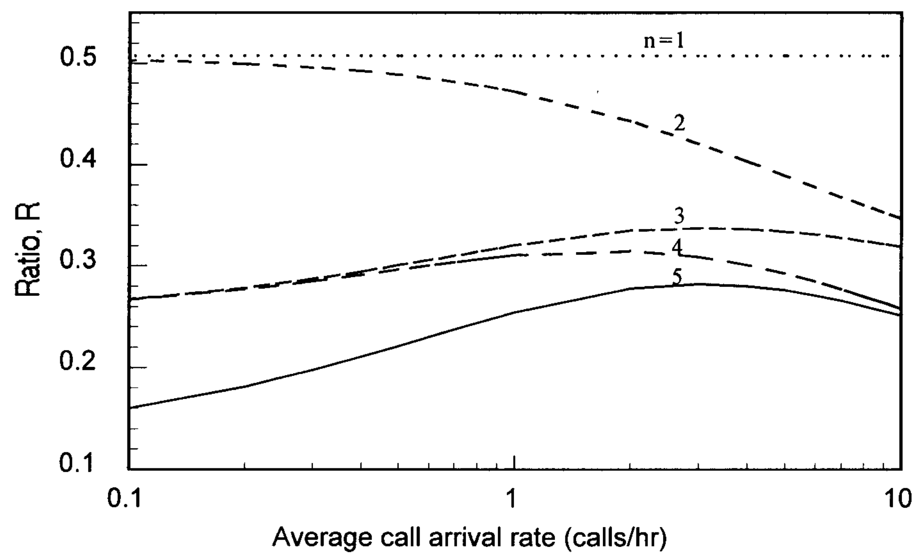


(b) Average call arrival rate, $\overline{\lambda} = 5$ calls/hr

Fig. 5: The relative average combined cost of a mobile user for the proposed scheme.
Parameters: N=1024.
For case of n=1, w=r=32.
For cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$.
For cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$.

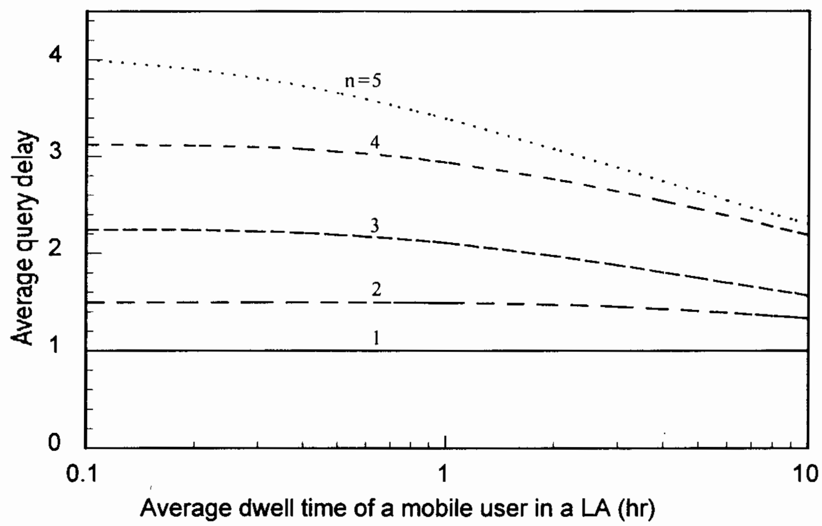


(a) Average dwell time, $\overline{T}_D = 0.2$ hr.

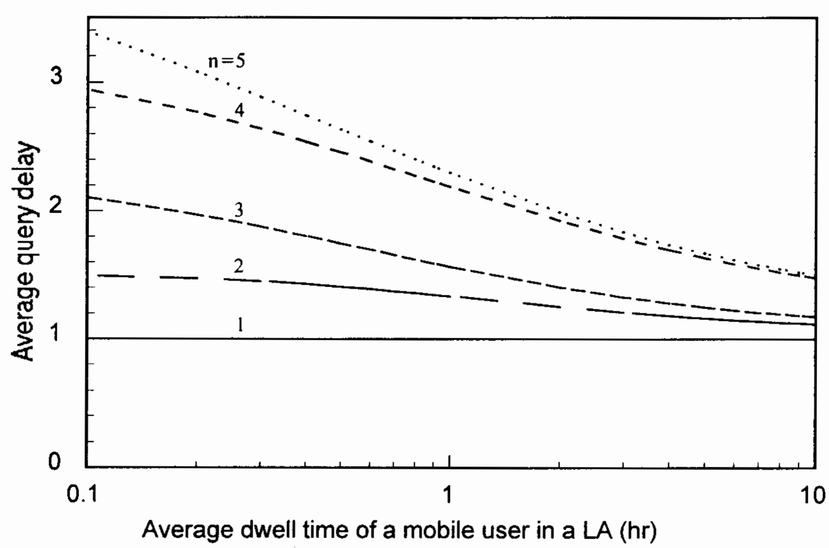


(b) Average dwell time, $\overline{T}_D = 2$ hr.

Fig. 6: The relative average combined cost of a mobile user for the proposed scheme.
Parameters: N=1024.
For case of n=1, w=r=32.
For cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$.
For cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$.



(a) Average call arrival rate, $\overline{\lambda} = 0.5$ calls/hr



(b) Average call arrival rate, $\overline{\lambda} = 5$ calls/hr

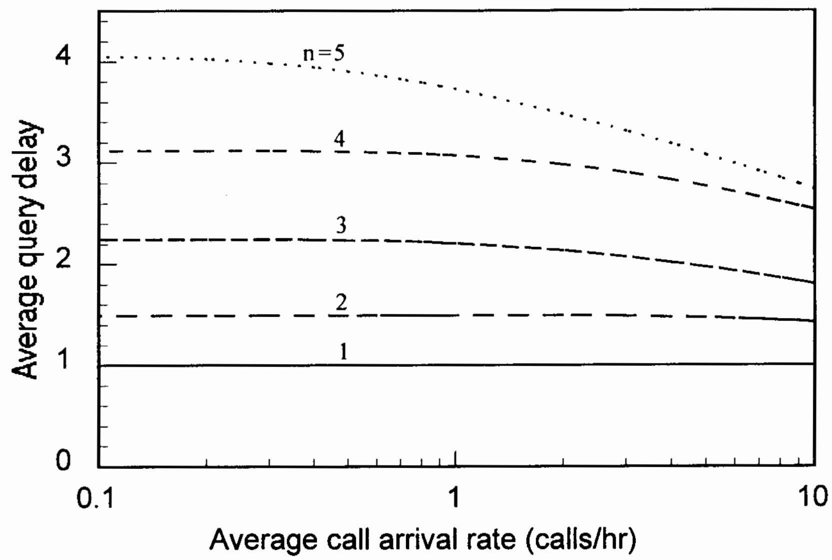
Fig. 7: Average query delay for the proposed scheme

Parameters: N=1024, $t_r = 1$.

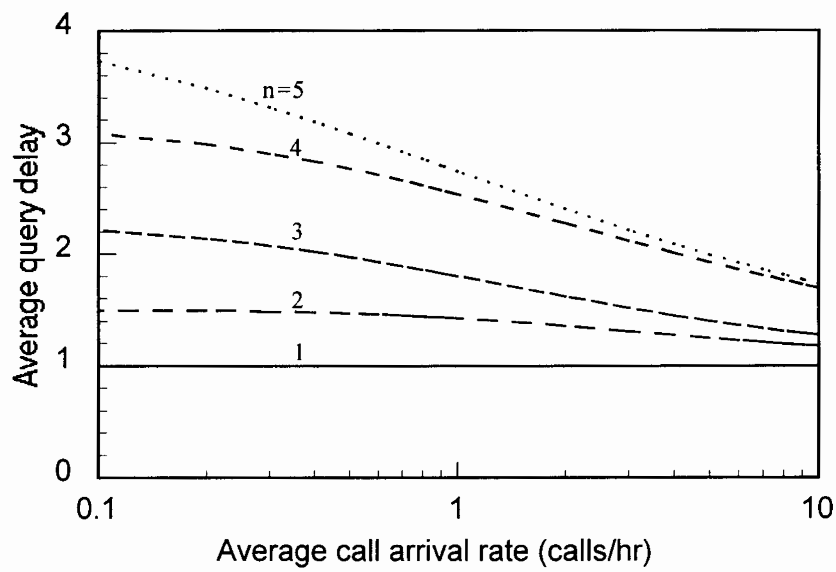
For case of n=1, w=r=32.

For cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$.

For cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$.



(a) Average dwell time, $\overline{T}_D = 0.2$ hr.



(b) Average dwell time, $\overline{T}_D = 2$ hr.

Fig. 8: Average query delay for the proposed scheme
Parameters: N=1024, $t_r = 1$.
For case of n=1, w=r=32.
For cases of n=2 and 3, $w_1 = 64$ and $r_1 = 16$.
For cases of n=4 and 5, $w_1 = 128$ and $r_1 = 8$.