Adaptive Location Prediction Strategies Based on a Hierarchical Network Model in Cellular Mobile Environment *

Sajal K. Das and Sanjoy K. Sen
Department of Computer Sciences
University of North Texas
Denton, TX 76203-6886
E-mail : {das,joy}@cs.unt.edu

Abstract

We present four efficient heuristics (one basic scheme and three of its variants) to predict the location of a mobile user in the cellular mobile environment. The proposed location management schemes assume a hierarchy of location areas, which might change dynamically with changing traffic patterns. A method to compute this hierarchical tree is also proposed. Depending on the profile of the user movements for the last τ time units, the most probable (and the future probable) location areas are computed for the user in the basic scheme and its first variant. The second variant predicts the location probabilities of the user in the future cells combining them with those already traversed in the last τ time units to form the most probable location area. The third variant is a hybrid of the first and second variants. Finally, the proposed heuristics are validated by extensive simulation of a real time cellular mobile system, where all the four schemes are compared under various traffic patterns.

1 Introduction

Location management is an important problem in the cellular mobile environment. Such an environment typically consists of a number of geometrical areas called *cells*, each having a *base station* (BS) interfacing with numerous *mobile stations or users.* A number of base stations may be joined by a *mobile switching center* (MSC) which also acts as the interface between the base stations and the existing backbone networks like PSTN, ISDN etc. Location management fundamentally deals with the problem of locating a mobile user at any time at any place in the given geographical area. A search information is broadcasted in a single (or multiple) cell(s) for the user. This is called *paging.*

Various types of location management schemes have been suggested by the researchers in the last few years [See reference]. We enumerate below some of the fundamental types of schemes that have been proposed in a scattered manner in the literature.

1. If the cell location of the user is known exactly, then any user is guaranteed to be found by paging a single cell. For this scheme to work, the current cell location for each user has to be stored in a database and updated correctly and efficiently.
2. At the other extreme, paging is done in the whole system area, which requires a broadcast call to each and every cell. As a result the amount of channel bandwidth consumed by these numerous broadcast calls for paging can be extremely high.
3. A more efficient scheme [TGM 88] will keep track of a *location area* (consisting of a cluster of cells) where the mobile user might be found, and use the paging scheme simultaneously in all the cells within that area. The aim is to minimize the total overhead in locating an user which comprises both the signalling overhead in paging the cells and the overhead due to updating the user locality.
4. Another type of location management scheme [XTG 93] uses the same location area based searching mechanism, but allows the location areas for each user to vary in size dynamically depending on the rate of call arrival. An optimal value of the size of location area for the user is computed based on minimizing the cost function, comprising of paging and updating cost components.
5. Another approach [BIV 92] partitions the location area into disjoint sets of cells based on a *user profile* (which is generally a history of user move-

---

*This work is supported by Texas Advanced Technology Program grant TATP-003594031, and by a grant from Bell Northern Research (BNR) Inc., Richardson, Texas.

ments) called partitions. Each user will be allocated a set of partitions of cells. The updation will take place only when the user changes partition.

6. A new class of location management scheme seeks to predict the current location of the user, based on a history of user movements and futuristic tendencies [Rok 90]. This is done by both the base-station and the mobile station, and compared (updated) from time to time.

Our location management schemes are prediction based. They are based on a hierarchical tree of dynamically changing location areas constituting the cellular mobile environment. First, an approach to construct this hierarchical tree using a newly introduced concept of the average egress rate (AER) of a location area, is described. Then, based on the history of user movements over a certain period of time, the location area in this hierarchy where the user is most likely to be found (most probable location area, MPLA), is computed. Searching takes place in MPLA based on an assignment of location probabilities for the user in the children location areas of the MPLA. This heuristic also computes the future probable location area of the user. The first variant of our basic scheme leads to an improvement of the tracking time for the user, using a certain threshold probability as an input parameter. The second variant tries to predict the traversal frequencies of the user in the future cells (cells not yet traversed but most likely to be in the not-so-distant future) and computes the most probable location area for the user based on its history of movements and the predicted future traversal frequencies. This scheme is expected to improve the latency of tracking down a mobile user. The third variation is a hybrid of the previous two variants and is expected to combine the advantages of the two. Detailed simulation experiments are carried out with various types of user movement data to validate the effectiveness of our schemes.

The rest of the paper is organized as follows. A new logical architecture of the cellular mobile system is proposed in Section 2. A new scheme is proposed in Section 3. Three variants of this scheme are described in Section 4 and simulation studies are described in Section 5. In Section 6, we briefly compare our schemes, qualitatively, with some of the existing work. Section 7 concludes the paper.

2 Logical Architecture of the Wireless System

The wireless system consists of a number of imaginary cells of predefined boundaries each with a base-station communicating with the mobile stations through wireless links. The cells form the lowest level in our hierarchical logical architecture as shown in Figure 1. Two or more cells are grouped into a *location area* (LA) which will form the next level of the hierarchy. The LA's again combine to form larger location areas in the next level of the hierarchy and so on. The grouping of the cells into LA's is governed by what we call the *average egress rate* (AER) from a cell or a location area, which we introduce in the next section. This logical hierarchy of LA's can change dynamically governed by the changing AER's of the cells due to changes in the traffic pattern.

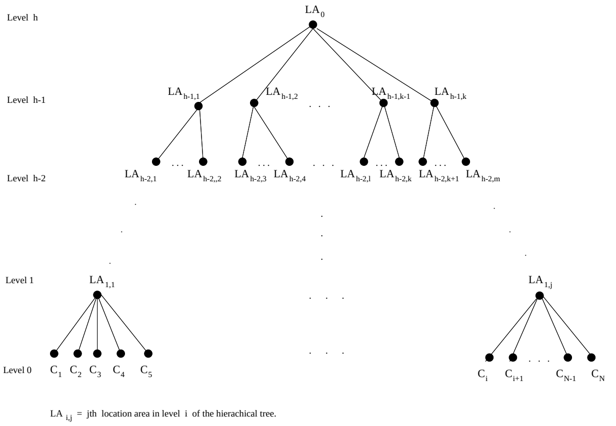

Figure 1: Logical architecture of a wireless system

A mobile switching center (MSC) is in charge of a cluster of cells, but this cluster may not be the same as an LA. In fact a cluster belonging to a particular MSC may consist of one or more LA or parts of the same. On the contrary, a particular LA may comprise of one or more MSC's or parts thereof. But we assume that, a particular mobile user is enrolled in a home register database at a particular MSC. There is a location server in that MSC which is responsible for updating the location prediction informations for the user and also initiating the search for the user when called for. There are several such location servers each responsible for one or more users, which might be running in a particular MSC.

3 A New Location Management Scheme

A most probable location area (MPLA) is assigned to each mobile user depending on the profile which it carries and maintains, and which it communicates to its location server every τ units of time. The location server maintains the user location database. Due to the power-saving considerations of the mobile terminal or equipment, all the major overheads involved in computation of the user location are done by the location server. The user only conveys the neccessary information. The main implication of this is that the amount of channel bandwidth wasted due to signalling will be negligible as far as location updates is concerned. The user does not need to do periodic registration or deregistration as in most of the other schemes. The user profile generally consists of four kinds of information: (*i*) the history of its movements in the past τ time units, *i.e.* a list $L_c = \{C_i, f_i\}$ of cells (C_i) traversed during that time along with the associated frequency (f_i) of traversals; (*ii*) the mean velocity until now; (*iii*) the direction of future movement (optional); and (*iv*) the last cell traversed.

The first information is useful in determining the current most probable location area of an user. If its movement in the last τ units of time is confined to a single cell, then that cell is its current location area. If the user has changed cells several times in that period, then from the user profile the cells traversed by it during that time is known, and the most probable location area assigned to the user will be the lowest common ancestor of these areas in the hierarchy of LA's. Once the most probable location area of the user is determined, it is required to trace the user in a particular cell of that location area. The system tries to predict the correct cell depending on the past user movements. According to the frequency of traversals of the cells in a particular location area, a probability of locating the user is assigned to each cell and then to the parent LA. A way to assign these probabilities is described in Subsection 3.1. Whenever the most probable LA for the user is known, a search has to be initiated in the various children LA's under that LA. The search is continued starting with that child LA having the highest probability for locating the user. Thus we apply a kind of depth-first search on the hierarchical tree of LAs everytime selecting the child LA with the highest location probability for the user. In this way, when the lowest level of the hierarchy (*i.e.*, a particular cell) is reached, that cell is paged. If the user is not found there, then we backtrack the search tree, and initiate the search in the subtree of the LA having the next lower probability of locating the user.

The location server of the user also tries to predict the next LA towards which the user is heading depending on the mean speed and direction or the future direction (if known) in the user profile. If the mean velocity of the user is v, the average distance expected to be covered by the user in the next τ time units is $v \cdot \tau$. Thus depending on the current or future direction of the user, it is easy to predict the cells which can be reached by the user starting from the last known cell, in the next τ units time window. These cells are called the *future cells.* The lowest common ancestor of those cells among the future cells *which do not belong to the current* MPLA, is the *future probable location area* (FPLA) of the user. The problem now is to assign a certain probability that the user will reach the FPLA, which is discussed in detail in the next subsection. The implication of the FPLA is that, if a search fails in the MPLA (most probable location area), then the search will be initiated next in the FPLA. However in FPLA the search is conducted simultaneously in all the cells under it. If the search also fails there, then a new search is initiated in the entire wireless system.

3.1 Computation of Location Probabilities

Let us first consider a specific example of the logical architecture of a wireless system in which the hierarchy of LA's at some part of the day is shown in Figure 2. There are ten cells, $\{C_1, C_2 \ldots C_{10}\}$ in the entire area such that cells C_1, C_2, C_3 are clustered to form a location area called LA_1. Similarly, cells C_4 and C_5 form LA_2, while cells C_6, C_7, C_8, C_9 form LA_3 and cell C_{10} itself forms LA_4. Next LA_1 and LA_2 together form the location area in the next level of the hierarchy, namely, LA_5. Similarly, LA_3 and LA_4 cluster to form LA_6. In the highest level of the hierarchy, LA_5 and LA_6 merge to form the entire location area, say LA_0.

The location server for the user is responsible for computing the location probability of the user at various location areas. We assume that every location server knows the current hierarchy (as in Figure 2) of LA's in the system. The mobile user can keep track of the number of times it has traversed a particular cell based on the periodic broadcast information from the base stations. The user maintains a list $L_c = \{C_i, f_i\}$ in its profile, where f_i is the frequency of traversals to the cell C_i in the last τ time units. The mobile user conveys these informations to its location server. From this information, the location server computes the location probabilities for the user in those areas, assuming that these probabilities in the next τ time

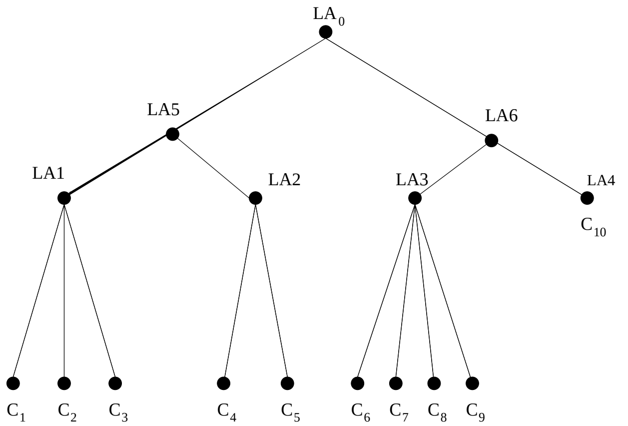

Figure 2: A specific example of the logical architecture of a wireless system

units will be directly proportional to these traversal frequencies, *i.e.* location probability of the user in cell C_i is $\frac{f_i}{\sum_i f_i}$. Once we know the location probablities of the cells, the location probabilities of the LA's in the higher levels of the hierarchy can be easily computed, until we reach the current most probable location area (MPLA), which will have a location probability of 1 for the user in this case. The future probable location area (FPLA) of the user can be predicted on the basis of its past movement and velocity. The way to predict this LA has already been discussed in course of describing the location management scheme. The probablity, P_f, of finding the user in the FPLA will depend on the pattern of user movement in the last τ time units. Based on the information $L_c = \{C_i, f_i\}$ we compute the ratio, $R = \frac{|L_c|\left(1-\frac{1}{1+\sum_i f_i}\right)}{\sum_i f_i}$. We claim that this ratio gives us a good estimate for P_f. The justification for this claim is as follows. Suppose the movement of the user is confined to a few cells but is highly repetitive in nature, then we can expect that the user will continue to remain confined within those few cells, *i.e.* it will have a low probability of moving to the future probable cell. In this case, the value of R will be low because a highly repititive movement implies a lot of changing of cells, increasing $\sum_i f_i$ consequently. If the movement of the user is confined to a few cells but the user does not change cells more frequently, then we expect that it will stay in the same location area or change location areas with almost equal probabilities. In this case, the factor $1 - \frac{1}{1+\sum_i f_i}$ determines the probability value, as $\frac{|L_c|}{\sum_i f_i}$ is close to 1. The third kind of movement the user might have is continuous motion in a particular direction (*e.g.* highway motion), which means that the user travels through many cells with a traversal frequency of one. This implies that R will have a value close to one signifying the fact that the user has a high probability of getting into the future probable LA.

3.2 Computation of Average Egress Rate and Location Areas

The *average egress rate* (AER) of a cell is defined as the average rate of outflow of the mobile users from that cell over a certain period of time. The AER of a location area (LA) consisting of more than one cell is the average AER of traffic in the boundary cells flowing out of the LA. A cluster is formed out of a group of cells such that the AER of the cluster is smaller than a certain threshold value. In this way, cells are dynamically grouped to form clusters, called LA's, with the intention of keeping the net outflow of traffic from each cluster as low as possible. Depending on the changing traffic pattern throughout the day, this clustering may change dynamically.

Assuming that the cells are square in shape, we now give an algorithm to compute the *level 1* location areas from these cells. Let the average outflow of traffic (i.e. average egress rate) through side k of cell C_i, is $AER_{i,k}$, where $k \in \{1,2,3,4\}$. The algorithm can be easily extended to cellular architectures with hexagonal cells, where $k \in \{1,2,3,4,5,6\}$. The proposed algorithm uses a modified breadth-first graph traversal technique to compute the first level LA's assuming that there is a pre-defined threshold value as the upper bound of the AER of any location area. A rectangular array of cells, $C = \{C_i\}$, constitutes the given wireless system.

```
Algorithm : Compute_LA
Step 1: For each cell $C_i$ compute $AER_{i,k}$, for $1 \le k \le 4$.
Step 2: Construct a graph $G = (V, E)$ from the given rectangular array of cells such that $V = C$ and $< i, j > \in E$ if the cells $C_i$ and $C_j$ have an edge or a point in common. Select a node $C_s$ as the starting node.
Step 3: $\mathcal{S} = \{C_s\}, AER_{\mathcal{S}} = \sum_{k=1}^{4} AER_{s,k}$.
Step 4: Run the algorithm Modified-BFS in the graph starting with the node $C_s$.
end Compute_LA

Algorithm : Modified-BFS
Let us call the node $C_s$ the starting node $s$.
Mark $s$;
Enqueue $s$; j := 1;
While queue not empty Do
    $u \leftarrow$ Head of queue;
```

```
    For each $v \in Adj(u)$ do
      if $v$ not colored then
        i) Let $U$ be the set of common edges with $v$
        of the non-colored or non-marked neighbors
        of $v$;
        Let $L$ be the set of edges of $u$ common to $v$.
        ii) Set $decrease := \sum_{l \in L} AER_{u,l}$,
        Set $increase := \sum_{i \in U} AER_{v,i}$;
        iii) if $AER_{\mathcal{S}} + increase - decrease \leq$
        $threshold$ then
          $AER_s := AER_s + increase - decrease$;
          $\mathcal{S} := \mathcal{S} \cup \{v\}$;
          mark $v$;
          Enqueue $v$;
        else
          $LA_{1,j} := \mathcal{S}; \mathcal{S} := \{v\}$; mark $v$;
          Enqueue $v$; j := j + 1;
        endif
      endif
    endfor
    Dequeue queue;
    color $v$;
endwhile
end Modified-BFS.
```

This algorithm is used iteratively to construct the LA's in every level of the hierarchical tree from the LA's in the previous level, until none of the remaining LA's can be further combined into a larger LA satisfying the property that its AER is less than the threshold value. These LA's are combined to give finally the whole cellular environment at the root of the tree.

The location management scheme is inherently sequential and herein lies its disadvantage. For example, if the user is actually located in one of the less probable cells, then the latency for tracking him down can be substantial. The first variant of the scheme proposes to overcome this disadvantage. Again, the algorithm performs very well for repetitive patterns of movement by the user. If the user tends to move in a straight direction, this prediction based scheme does not perform well. The second variant tries to make sensible predictions for future locations of the user.

4 Three Variants of the Location Management Scheme

4.1 Variant I

Instead of paging each cell individually, this first variant of our scheme seeks to page a location area consisting of more than one cell, simultaneously. That location area might be the most probable location area for the user or a descendant in the hierarchical subtree of LA's with the most probable location area (MPLA) as the root. The choice of the location area to be paged will depend on the probability of locating the user in that LA. We compute a certain threshold probability P in terms of the maximum allowable increase in paging cost and the maximum value of a function constituting the number of cells and the user location probabilities in those cells under the LA's, which belong to the MPLA.

Our modified scheme to track down the user is as follows. As we traverse the hierarchical subtree of the LA's with the most probable LA as the root, in the way described in Section 3, we traverse the LA's in the decreasing order of probabilities. This is because the probabilities of the children LA's are summed up to form the location probability of the parents. As the tree is traversed in the depth-first order, the LA with the maximum depth from the root of the subtree with location probability greater than or equal to the threshold probability, P, is chosen. The entire LA is paged simultaneously. If the user could not be traced in this area, the remaining cells of the MPLA are sequentially paged in the order of decreasing probabilities. If the user is not found there, then the FPLA of the user is searched. And if the user is still not found there, the entire system wide paging is resorted to.

4.1.1 Performance Analysis

Let the most probable LA of a certain user be $LA_{i,j} = \{C_n, C_{n+1}, \ldots C_{n+l-1}, C_{n+l}, \ldots, C_{n+m-2}, C_{n+m-1}\}$. Let $LA_{k,l}$ be the descendent of $LA_{i,j}$ which is selected for paging the user. Without loss of generality, let us assume that $LA_{k,l} = \{C_n, C_{n+1}, \ldots, C_{n+l-1}\}$, a subset of $LA_{i,j}$ containing the first l cells. These l cells are paged simultaneously for the user. Let the probability of locating the user in $LA_{k,l}$ be $P_s \geq P$ (threshold probability). The total cost of paging in this scheme is, $PC(2) = lC_p + (1 - P_s)\{1 + \sum_{j=n+l}^{n+m-1} \prod_{i=0}^{j-(n+l)} (1 - P_{n+l+i}) + N(1 - P_f) \prod_{j=n+l}^{n+m-1} (1 - P_j)\}C_p$. The increase in paging cost due to this variant over the basic scheme (Section 3) is given by,

$$\triangle PC = \{l - \sum_{j=n}^{n+l-1} \prod_{i=0}^{j-n} (1 - P_{n+i}) - P_s\}C_p.$$

Now suppose a certain threshold value for $\triangle PC$ is given beyond which we cannot increase the paging cost. Call this ΔPC_{max} . This will help us determine the value of the threshold probability, P, for a certain allowable range of variation of ΔPC_{max}, as follows.

Step 1: Considering all the internal nodes of the hierarchical sub-tree of LA's with the MPLA as the root, find out the maximum value of the function,

$l - \sum_{j=n}^{n+l-1} \prod_{i=0}^{j-n} (1 - P_{n+i})$, where l is the number of cells which belong to the LA. This is easy to find because we know all the location probabilities for the user in all the cells and LA's under the MPLA. Call this maximum value Γ_{max}.
Step 2: Set $P = \Gamma_{max} - \frac{\Delta PC_{max}}{C_p}$ so that the allowable range of ΔPC_{max} satisfies $(\Gamma_{max} - 1) \cdot C_p \leq \Delta PC_{max} \leq \Gamma_{max} \cdot C_p$.

Obviously, in this variant of the scheme the latency of tracking a user is less than in the original scheme under similar conditions, although the paging cost is generally more. But the choice of the threshold probabilty takes care that this increase in paging cost never exceeds the allowable limit.

4.2 Variant II

A major drawback of our basic location management scheme or its variant I is we cannot assign any frequency to those cells which are not yet traversed by the user in a τ units of time window which consequently have a zero location probability for the user in the next τ time units, although they may lie just ahead in its trajectory and in reality have a high probability of visit by the user in the near future. Another variant of our location management scheme is in the way we assign predictive traversal frequencies to the so called future cells which may belong to the MPLA or to the FPLA. In our original scheme, we always assign a zero probability to the cells not yet traversed by the user. Although we do assign a certain probability to the FPLA, we do not associate any probability to the cells belonging to that LA. Hence the entire future probable LA needs to be simultaneously paged. In the Variant II, first we need to determine from the user profile data, the cells most likely to be traversed by the user in the next τ units of time window. Knowing the mean velocity v of the user, the cells lying in the direction of movement of the user within a range $v \cdot \tau$ from the cell last traversed, are known as the *future cells* (see Section 3.1). Next we assign a certain predicted frequency of future user traversal to these cells. This frequency is given by $R \cdot (max\{f_i\})$, where $R = \frac{|L_c|\left(1 - \frac{1}{1 + \sum_i f_i}\right)}{\sum_i f_i}$ (see Section 3.1) and $max\{f_i\}$ gives the maximum frequency of traversal of any cell in the last τ time units. The region of search for the user now includes not only the cells traversed by the user in the last τ units of time window, but also the future probable cells expected to be visited in the next τ units time window. The current MPLA now becomes the lowest common ancestor of all these cells, and search for the user now proceeds exactly as laid down in the basic scheme. The pseudocode of the algorithm for Variant II is given below. It consists of two main parts:

```
Algorithm : Location_Manager()
Step 1: From the user_profile, compute the future probability $R$. Also find the maximum traversal frequency.
Step 2: Compute the future cells.
Step 3: Assign the predicted traversal frequency (product of $R$ and the maximum traversal frequency) to the future cells.
Step 4: Compute the MPLA consisting of the cells in the user profile as well as the future cells.
Step 5: Compute the location probabilty for the user in all the cells and LA's under the MPLA.
end Location_Manager.
Algorithm : Location_Finder()
Step 1: Do a depth first search of the hierarchical subtree with MPLA as the root traversing the cells and the LA's in the decreasing order of probabilities. If the user is found in one of the cells, return.
Step 2: Page all the cells of the system simultaneously.
end Location_Finder.
```

In Variant II, if the probabilities in the yet-to-be traversed cells can be properly estimated, then the search for the user might begin in one of the future probable cells when his motion is uniform in a particular direction. Due to the immense likelihood that the user might be found in one of these future cells, in that case, this scheme is expected to produce less latency for tracking down the mobile user. Next we show by an example how this new scheme works. Let us consider the case where we want to track an user whose profile for the last τ time units is known. The part of the profile showing the recorded traversal frequencies for the cells $\{C_1, C_2, \ldots, C_8\}$ is shown in Figure 3. The current MPLA for the user is determined as that

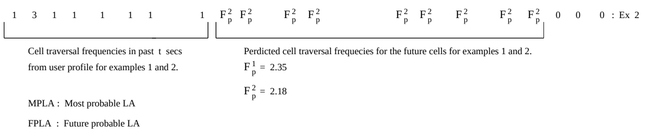

Figure 3: An example scenario demonstrating the basic scheme and its variation

LA which is the lowest common ancestor of the cells $C_1 - C_8$. In this case it is LA_{15}. Also, from the velocity information in the profile, we have computed the future probable cells $C_9 - C_{17}$, *i.e.* cells which lie in the current trajectory of the user within a range $v \cdot \tau$ from the last recorded cell, which we assume is C_8. Among these cells those which do not belong to the MPLA, form the FPLA. In this case, cells $C_{13}, \ldots, C_{17}$ form the FPLA, denoted as LA_{14}.

To make the computation of location probabilities simpler, we assume that all the cells are disjoint. From our basic location management scheme (Section 3), the computation of the location probabilities is done as follows. All the future cells are allocated a traversal frequency of zero for the user. The location probability for each cell C_j in the MPLA for the user is now computed as $P_{C_j} = \frac{f_j}{\sum_i f_i}$, where f_i is the traversal frequency for the cell C_i in the MPLA. In this way, we get $P_{C_1} = \frac{1}{16}, P_{C_2} = \frac{5}{16}, P_{C_3} = \frac{3}{16}, P_{C_4} = \frac{1}{16}, P_{C_5} = \frac{2}{16}, P_{C_6} = \frac{1}{16}, P_{C_8} = \frac{1}{16}, P_{C_9} = 0, P_{C_{10}} = 0, P_{C_{11}} = 0, P_{C_{12}} = 0$.

Next we compute the location probabilities of the LA's in the next level of hierarchy. We get $P_{LA1} = \frac{10}{16}, P_{LA_2} = \frac{2}{16}, P_{LA_3} = \frac{3}{16}, P_{LA_4} = \frac{1}{16}, P_{LA_5} = 0$. For the next higher level, $P_{LA_{10}} = \frac{15}{16}, P_{LA_{11}} = 0, P_{LA_{12}} = \frac{1}{16}$ which add up to 1 for the MPLA (LA_{15}). Then the value of R is computed as $R = \frac{8}{16}(1 - \frac{1}{17}) = \frac{8}{17}$ which is the location probability assigned to the FPLA. Now the search takes place starting from the cell C_2 which has the highest location probability for the user.

Let us consider what happens in the case of variant II. For our example, first we compute $F_p = R \cdot max_i\{f_i\} = \frac{8}{17} \times 5 = 2.35$. Assign F_p as the predicted frequency of all the future cells, assign zero frequency to all other cells belonging to the MPLA and the FPLA. Next the location probabilities for the user in all these cells are computed as: $P_{C_1} = \frac{1}{37.15}, P_{C_2} = \frac{5}{37.15}, P_{C_3} = \frac{3}{37.15}, P_{C_4} = \frac{1}{37.15}$. Hence $P_{LA_1} = \frac{10}{37.15}$. Similarly, $P_{LA_2} = \frac{2}{37.15}, P_{LA_3} = \frac{3}{37.15}, P_{LA_4} = \frac{5.7}{37.15}, P_{LA_5} = \frac{4.7}{37.15}, P_{LA_6} = \frac{4.7}{37.15}, P_{LA_7} = \frac{2.35}{37.15}, P_{LA_8} = \frac{4.7}{37.15}, P_{LA_9} = 0$. Similarly, $P_{LA_{10}} = \frac{15}{37.15}, P_{LA_{11}} = 0, P_{LA_{12}} = \frac{10.4}{37.15}, P_{LA_{13}} = 0, P_{LA_{14}} = \frac{11}{37.15}$, and so on. Note that the probability of locating the user in LA_{10} has decreased from $\frac{15}{16}$ to $\frac{15}{37.15}$, while that in the future cells has increased. But the search for the user starts in the cells already traversed in the last τ time units before going to the future cells because $P_{LA_{10}}$ is still greater than $P_{LA_{12}}$ (LA_{12} contains most of the future cells). From the pattern of the movements of the user, this is not unreasonable. The next example shows what happens when the user movement patterns definitely predict a very high possibility that the user will be found in any of the future cells in the next τ units.

The cells traversed by the user along with the corresponding frequency of traversal is shown in the Figure 3 in the entry denoted by Ex 2. Following similar calculations as shown earlier, it is found that for our basic scheme, $P_{LA_{10}} = \frac{10}{11}$ and $P_{LA_{12}} = \frac{1}{11}$, while for the variant II, the same probabilities come out to be $\frac{10}{35.09}$ and $\frac{11.68}{35.09}$ respectively. Hence, for this variation of the basic scheme the search starts in the cells under LA_{12} (which mainly consists of the future cells) instead of in LA_{10} (which consists of the past cells traversed by the user). This demonstrates the efficacy of this variant scheme to give a good prediction of the future location of the user depending on its past history of movement.

4.3 Variant III

The variant II of our location management scheme suffers from the same drawback as the basic scheme. It is inherently sequential in nature and hence in case of bad predictions the average tracking time for a user will be high. A logical improvement suggests that a combination of the Variants I and II of the basic scheme be used as the hybrid location management strategy. This will combine the good prediction capabilities of Variant II with the lower average latency of the scheme in Variant I, and is expected to perform well under all sorts of traffic conditions.

5 Simulation Results

Detailed simulation experiments are carried out for our prediction-based location management scheme and its three variants. Experiments are conducted under various types of user movement patterns, noting how each location management scheme performs under all circumstances. After a brief discussion of the simulation environment, we summarize the results of the experiments.

5.1 Simulation Environment

For the purpose of simulation, we assume that the cellular environment consisted of a rectangular array of square cells. Let *numcells* denote the total number of cells in the array. Our simulation program consisted of four main processes: *location_server*(), *user*(), *setup_LA*() and *callee*(). The location_server() interacts with all the other three processes. It gets the *user_profile* from the user() at intervals of every τ time-units and computes the MPLA, location probabilities of the user, the FPLA etc. (as laid down in the discussions on the heuristics). It also receives the updated hierarchical tree of LA's at intervals of T units from the setup_LA(). It initiates a search for the user as soon as the callee() process asks it to do so.

The user() process keeps track of its current cell and cell changes, and also its average velocity. The mean speed is computed at each iteration from a pair of random variables denoting its current cell. The various patterns of user movement is controlled by the way this random variable is generated. We choose three different patterns of user movements as given below:

1. **Localized**, which means that the user movement is very repetitive and mainly concentrated in a very few cells.
2. **Not-so-localized**, which means that the number of cells that the user traverses (more or less randomly) is more than the localized pattern.
3. **Almost uniform**, which means that the user visits a large number of cells with almost equal frequency which is close to one. This is an approximation of the user travelling down a highway.

The user sends the user_profile data to the location_server() every τ time units. The contents of the user_profile has been described in details in Section 3.

The setup_LA() process reconfigures the location area hierarchy every T time-units based on the new information regarding the AER's of the cells, using the modified-BFS algorithm as described in Section 3. After recomputing the new LA's, it sends information to the location_server regarding the new LA tree. The callee() process only requests the location_server to track down the mobile user, the request being generated at random.

5.2 Emperical Performance

The simulation experiments are carried out for all the four schemes (i.e. the basic scheme and its three variations), under the above three different types of user movement patterns. The two performance metrics used to compare the efficacies of the schemes are (*i*) the paging cost, i.e. the number of cells paged to locate the actual cell the user is in, and (*ii*) latency, i.e. the search time. The following table (Table 1) compares the performance of all the four heuristics under the different patterns of user movements in one snap-shot of the experiment, where, Heuristic 1 = basic scheme, Heuristic 2 = Variant I, Heuristic 3 = Variant II, Heuristic 4 = Variant III. Also, * - paging cost = number of cells paged and ** - latency (tracking time), assuming average paging time for a cell is 2 msecs.

As expected the basic heuristic (heuristic 1) performs very well for localized user movement patterns.

Table 1: **Paging cost and latency for various patterns of user movements**

Scheme	Localized		Not-so-localized		Almost uniform	
	PC*	L**	PC*	L**	PC*	L**
Heuristic 1	3	6 ms	28	56 ms	14	28 ms
Heuristic 2	4	2 ms	28	44 ms	14	16 ms
Heuristic 3	3	6 ms	13	26 ms	8	16 ms
Heuristic 4	4	2 ms	14	2 ms	8	4 ms

It has the lowest paging cost (i.e. the number of cells paged) among all the heuristics presented. But the search being essentially sequential, the latency is high compared to its first and third variants, which employ simultaneous paging in multiple cells. But compared to other types of user movement patterns, however, this latency is much less. Variant I (heuristic 2) employs simultaneous paging in certain number of cells if the threshold probability condition is met. As a result there might be some redundant paging, but the latency is expected to be less than or equal to the basic scheme. From our experimental results, we see that only one extra cell is paged but the latency is much reduced compared to that of heuristic 1. Variant II (heuristic 3) in this case performs the same as the basic scheme, while the Variant III (heuristic 4) performs as well as heuristic 2. For not-so-localized user movement pattern, both the paging cost and the latency goes up substantially. This is expected, because our scheme is basically predictive in nature, which tries to predict the future location of the user based on its past movements. When the movement of the user is not repetitive, the performance of the heuristics is expected to degrade. However, in a real-life environment, mobile users with totally random patterns of movement is not very common! Heuristic 4 gives good performance as far as latency is concerned. Also in terms of paging cost, only heuristic 3 performs better than heuristic 4 by a small margin of one extra cell-paging.

For almost uniform motion, the performance of all the schemes are much better, although not as good as that for localized movements. Again heuristic 4 outperforms all other schemes. Except for the localized user movement patterns, the performance of the heuristics 2, 3 and 4 shows a general improvement from the basic scheme and also from each other, in that order.

The heuristics are exhaustively tested under five different user_profiles for each type of movement patterns, namely, localized, not-so-localized and almost uniform. The performance of the heuristics are depicted graphically (Figures 4 - 8). The first set of three graphs (Figures 4, 5) compare the paging cost when the four heuristics are run on the same user_profile data. It

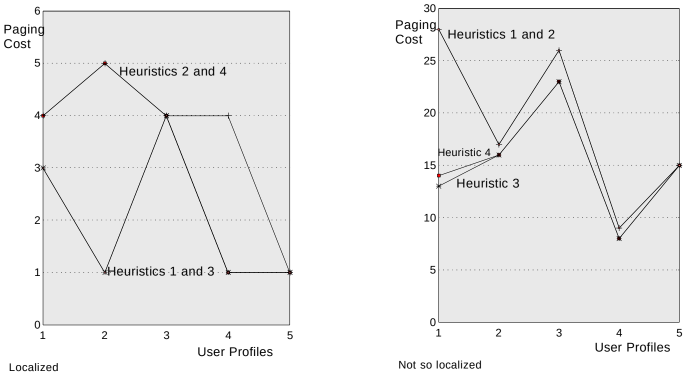

Figure 4: Paging costs for the heuristics with various user_profiles

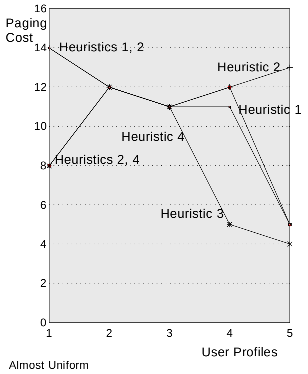

Figure 5: Paging costs for the heuristics with various user_profiles

can be seen that, except for the case of *localized* user movements, in the case of other two traffic patterns, the paging cost incurred by the Heuristics 3 and 4 is less than that of 1 and 2. However, considering all the cases, heuristic 3 performs the best (i.e. incurrs the least paging cost). If we consider the overall latency as the performance metric (Figure 7), we see that for the same sets of user_profile data, the heuristics 2 and 4 always perform better than heuristics 1 and 3 (i.e. give lower latencies to find an user). Heuristic 4 performs the best in all cases. So, we may safely conclude that, under various traffic patterns and considering both paging costs and latencies as the performance metrics, heuristic 4 gives the overall best performance, as was expected.

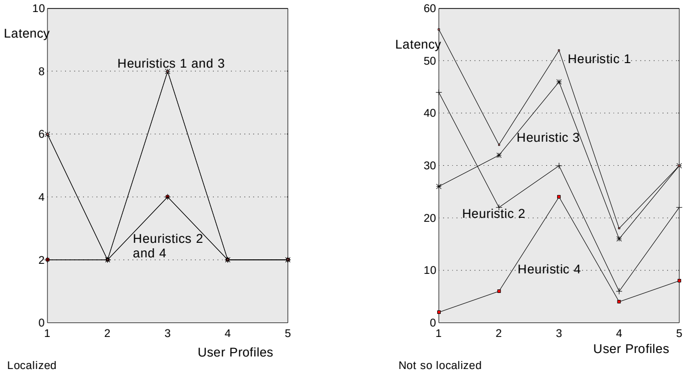

Figure 6: Latency for the heuristics with various user_profiles

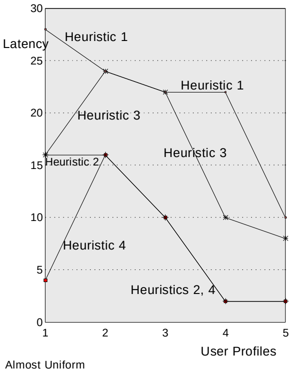

Figure 7: Latency for the heuristics with various user_profiles

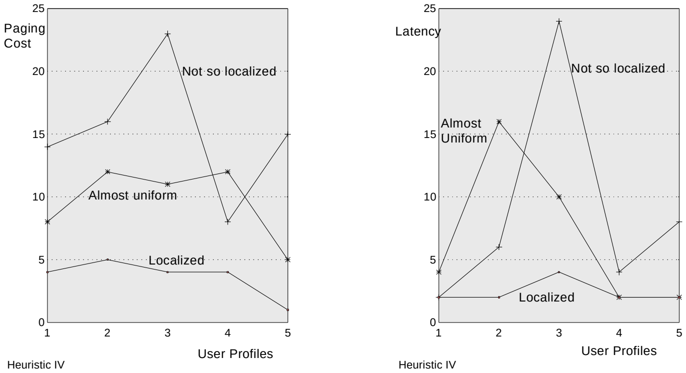

Figure 8: Performance of heuristic 4 under various traffic patterns

6 Comparision of Our Scheme With Some Related Work

Xie, Tabbane and Goodman [XTG 93] proposed a dynamic location area management scheme, where the location area for each user can change dynamically depending on the changing mobility patterns of the user. The optimal size of a location area for a user is obtained by minimizing a cost function constituting the cost of paging and the cost of update. In our schemes, the mobile user does not need to update even if it changes its current location area. Updation is based on the periodic sending of the user profile. Thus the amount of signalling traffic due to updates does not depend on the movement pattern of the user.

Badrinath, Imielinski and Virmani [BIV 92] suggest the scheme of partitions of the cells to reduce the amount of location update traffic. Each user is allocated a set of partitions depending on its movement profile. But the maintenance and updation of these partitions for each and every user is very computation intensive. Our scheme divides the entire wireless system into a hierarchy of global partitions called location areas (LA). Depending on the user movement profile, each user is allocated one of these LA's as the most probable location area (MPLA). Maintaining a global set of partitions is much less computationally intensive than maintaining a set of partitions for each user.

In the location updating technique called MULTI [OOYM 91], Okasaka, Onoe, Yasuda and Maebara proposed a method using multiple layers of location areas. The chief motivation behind this scheme is to reduce the amount of location update traffic which is concentrated in the bordering cells of a location area in order that the traffic distribution is uniform. In our schemes, the location update traffic consists in the periodic transfer of the user_profile by all the users and hence its distribution is independent of the cell position.

Madhavapeddy, Basu and Roberts [MBR 94] proposed an algorithm to compute the optimal paging zone for the user based on the probabilities of finding the user in various cells. To obtain this, Madhavapeddy et. al. suggested the use of the *location accuracy matrix.* The drawback of this scheme is that it tries to estimate the location probability of a particular user in a cell based on the number of entries of all users to that cell. This means that, even though a user has never entered a cell, it can have a very high probability of being found in that cell, depending on the number of other users (having the same last cell of registration) who have entered that particular cell. In our schemes, the location probability of the user in a particular cell is computed from its traversal frequency in that cell in the near past and hence is more accurate.

7 Conclusion

In this paper, we have presented four heuristics (a basic scheme and its three variants) for location management and prediction of the mobile users in a cellular mobile environment, which are expected to perform well under most common types of traffic patterns. First a hierarchical network model is proposed and a scheme to develop the hierarchy using a newly introduced concept called the average egress rate (AER) is presented. Next the basic scheme for predicting the location of the user is described. Our scheme keeps the location update signalling traffic very low as the users only need to communicate their profile periodically to their location server. The basic scheme suffers from some drawbacks, which are taken care of by the three variations of this scheme which are proposed next. The performance of the heuristics are tested by extensive simulation under three types of user movement patterns. The performance shows a general improvement from the basic scheme and its variations. Our schemes are next compared qualitatively with some of the major existing work in this area.

References

[TGM 88] R. Thomas, H. Gilbert, G. Maziotto, "Influence of the moving of the mobile stations on the performance of a radio mobile cellular network", *Proc. of the Third Nordic Seminar on Digital Land Mobile Radio Communications*, Sept. 1988.

[XTG 93] H. Xie, S. Tabbane, D. Goodman, "Dynamic location area management and performance analysis", *43rd IEEE Vehicular Technology Conference*, May 1993.

[OOYM 91] S. Okasaka, S. Onoe, S. Yasuda, A. Maebara, "A new location updating method for digital cellular systems", *41st IEEE Vehicular Technology Conference*, May 1991.

[Plass 94] D. Plassmann, "Location management strategies for mobile cellular networks of 3rd generation", *Proc. of the 44th Vehicular Technology Conference*, June 1994.

[BIV 92] B. R. Badrinath, T. Imielinski, A. Virmani, "Locating strategies for personal communication networks", Workshop on Networking of Personal Communications Appliances, Dec, 1992.

[MBR 94] S. Madhavapeddy, K. Basu, A. Roberts, "Adaptive paging algorithms for cellular systems", *Technical Report*, Bell Northern Research, Richardson, Tx, 1994.

[Rok 90] C.H. Rokitansky, "Knowledge-based routing strategies for large mobile networks with rapidly changing topology", *Proc. ICCC '90*, New Delhi, India, Nov. 1990.