User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems

Erdal Cayirci, *Member*, *IEEE*, and Ian F. Akyildiz, *Fellow*, *IEEE*

Abstract—The user mobility pattern (UMP) scheme is introduced for location update and paging in wireless systems where mobile terminals (MTs) maintain their history data in a database called user mobility history (UMH). During a location update, a UMP is derived from UMH and registered to the network. Unless the MT detects that it has moved out of the registered UMP, it does not perform any other location update. On the other hand, cells are paged selectively according to the cell entry times in the registered UMP upon a call arrival for the MT. The related data structures and the protocols for the UMP scheme are presented in the paper. The experimental results show that the UMP scheme outperforms the time-based and movement-based location update schemes as well as the blanket, selective, and velocity paging schemes.

Index Terms—User mobility pattern, user mobility profile, location management, location update, registration, regularity, paging, cellular wireless systems.

1 Introduction

The location management in wireless systems involves two tasks: *location update* and *paging*. The location information about a mobile terminal (MT) is maintained by the location updates. In current systems, an MT reports its location whenever it enters a new location area (LA). Since an LA consists of a number of cells, the exact location of an MT should be determined for a call delivery. This is done by paging the cells in the last registered LA.

When an LA is comprised of a group of cells that are permanently assigned to that LA, and is fixed for all MTs, the location management scheme is called *static*. The dynamic location management techniques are more adaptive to the mobility characteristics of MTs. They allow dynamic selection of the location update parameters and reduce the signaling traffic due to location management. The *time-based*, *movement-based*, and *distance-based location update techniques* are well-known dynamic schemes [1], [2], [3], [4], [9], [11], [12], [15], [24]. In the *time-based* technique [4], [15], an MT performs location updates periodically at a predefined time interval. In the *movement-based* technique [3], [4], [11], location updates are initialized after crossing a certain number of cell boundaries. The *distance-based* location update [4], [9], [12], [24] is performed when the distance from the last registered cell exceeds a predefined value.

There are also other dynamic location update techniques in literature. In [7], the *direction-based* scheme is proposed where an MT reports its location only when its moving direction changes. In [17], the *selective* location update technique is introduced where location update is not performed in every LA, and some LAs are skipped based on the transition probabilities and the cell dwell times. The *state-based* location update technique [14], where an MT decides to update its location based on its current state, is another dynamic technique. The policies, which are the combination of the known techniques such as the time and distance-based schemes, are proposed in [6], [13].

Another group of the location update schemes is based on user profiles or history data. In the *alternative strategy* [19], [21], MTs are tracked by utilizing a mobility pattern for each MT. In these patterns, a list of LAs ordered according to the location probabilities of the MTs are maintained. In [18], a multilayered model is developed to provide a multidimensional perspective view of mobility patterns. In [10], a scheme is proposed that determines the size and the shape of the LAs for each user dynamically on the basis of gathered statistics and incoming call patterns. In [8], the *movement-based* location update scheme is enhanced by maintaining the recent movement history of the MT. In the *LeZi* update [5] scheme, the movement history of the MT is sent to the network. The network constructs a search tree by using the registered history data. When a call arrives, a selective paging based on this search tree is carried out to determine the exact location of the MT.

The primary goal of location update is to reduce the paging cost, and the performance of the paging strategies are closely related to the location update scheme. The location update schemes must provide the network with enough information such that the paging cost is reduced under a given delay requirement. There is a tradeoff between the paging cost and the paging delay as well as between the paging and update costs. As the resolution of the updated location information increases, the number of cells to be paged decreases. Similarly, the longer the paging delays are, the lower the number of paged cells becomes.

The least paging delay is guaranteed by the *blanket polling* technique [23] where all cells in an LA are simultaneously paged upon a call arrival. This scheme has

- *E. Cayirci is with Harp Akademileri, Yeni Levent, Istanbul 80625, Turkey. E-mail: cayirci@cs.itu.edu.tr.*
- *I.F. Akyildiz is with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: ian@ece.gatech.edu.*

Manuscript received 11 Dec. 2001; revised 21 May 2002; accepted 27 Aug. 2002.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 6-122001.

high paging cost. The *selective paging* [16], [22], [25] is an alternative to the *blanket polling* [23] and reduces the paging cost, but increases the paging delay. In the selective paging, the location of the MT is predicted based on its location probability, and cells are paged sequentially starting from the cells where the MT is most likely to be present. Several other paging strategies such as the *shortest-distance first* [23] and *velocity paging* [2], [20] are also proposed. In the velocity paging, the system calculates the maximum distance that a mobile can travel based on its average velocity and the last registration time, and pages the cells, which are within this distance from the last registered cell. The paging schemes with delay bounds as a QoS constraint are introduced in [22] where the location areas are partitioned into clusters. The number of clusters in a location area guarantees that the system can page them sequentially without exceeding the given delay bound.

In this paper, we introduce the user mobility pattern (UMP) scheme for the location update and paging where MTs keep track of their UMPs in a data structure called User Mobility History (UMH). During a location update, an MT derives the most expected UMP from its UMH and registers it to the network. The registered UMP includes the cells expected to be visited and the predicted cell entry times for each of the cells in the UMP. Unless the MT detects that it has moved out of the registed UMP, it does not perform any location update. When a call arrives for an MT, it is paged in one cell at a time starting from the cell where it is expected to be according to the registered UMP. Although, MTs register a dynamic location area that consists of several cells, paging only one cell is generally enough to locate them. We show that the UMP scheme reduces the control-signaling traffic due to location management by several orders of magnitude comparing to the *time-based* [4] and *movement-based* [3] location update, and the *blanket* [23], *selective* [16], and *velocity* [2] paging schemes.

Our scheme differs from the other user profiles or history data based location update techniques [5], [8], [10], [18], [19], [21] in the following aspects:

- In our scheme, MTs are responsible for predicting and registering the UMPs. This approach reduces the signaling traffic for maintaining UMH, and increases the resolution and the accuracy of the data in UMH because MTs can track every cell that they enter without any need for extra signaling.
- A UMP is a sequence of pairs consisting of cell identification and expected cell entry times. These pairs are called UMP nodes. Since both the locations and the expected entry times are predicted in UMPs, an effective *selective paging* can be executed based on the call delivery times by using them. Maintaining the expected cell entry times for UMPs also helps to make more accurate mobility pattern predictions as explained in Section 2.

The remainder of this paper is organized as follows: In Section 2, we explain the UMP scheme. This section also addresses the details related to the system components like the structure of a UMH and the update procedure for a UMH. In Section 3, we describe the UMP-based location scheme and discuss the details of the UMP-based paging scheme in Section 4. We develop mathematical models to examine the accuracy of UMPs and the registration/paging costs in Section 5. We evaluate the performance of the developed schemes by several experiments in Section 6. Finally, we conclude the paper in Section 7.

2 The User Mobility Pattern Scheme

Mobile subscribers usually follow a limited number of mobility patterns in their daily lives. For example, people generally take almost the same path and same time to go to work every day. In the UMP scheme, an MT collects the data related to these patterns in UMH, and predicts the UMP based on the collected data. During location updates, the expected UMP is registered to the network.

2.1 User Mobility Pattern (UMP)

A UMP is a list of cells expected to be visited starting from a given time according to the mobility history of a mobile, and is made up of a number of nodes that have two fields, namely, *cell identification* (cell id) and *expected cell entry time*. A *cell id* and *expected cell entry time* pair identifies a node, which is unique in a UMP. However, a UMP may have two different nodes with the same *cell id* and different *expected cell entry time* values. Similarly, two different UMPs may have nodes with the same cell id because the same cell may be visited in multiple UMPs or multiple times in the same UMP. For example, one may go to work passing through a cell in the morning. The user may come back home passing through the same cell in the evening. Morever, the same person may go somewhere else through the same cell at noon in the weekends. In these cases, an MT visits the same cell in different UMPs or in the different phases of the same UMP, i.e., the subscriber may have a single UMP that represents his way both to and from his work where some cells are visited twice at different times within the same UMP.

In Fig. 1, we show an example UMP with four nodes. The cell in the first node is expected to be entered starting from 1,080 according to the UMH of the MT that registered this UMP. Expected cell entry time is the representation of daytime in minutes with a 24-hour scale starting at midnight, i.e., 06:00 p.m. is represented as 1,080. Cell 12 is expected to be visited twice in Nodes 2 and 4 according to the UMP shown in Fig. 1. The difference between these two nodes is the *expected cell entry times*. Assuming that this UMP represents the pattern that a subscriber follows to go to work and to come back home, that user passes through Cell 12 once in the morning and once in the evening. Note that a UMP is valid after its registration as long as another UMP is not registered, which means that a UMP can span many days. For instance, a subscriber may go to work and nowhere else during some days where he does not need another location update when the UMP representing the path to work is registered before in the beginning of this period.

2.2 User Mobility History (UMH)

UMPs are derived from UMH which is a data structure where the mobility history of an MT is stored. Each MT

Node 1	cell id	8
	expected cell entry time	1080
Node 2	cell id	12
	expected cell entry time	480
Node 3	cell id	21
	expected cell entry time	510
Node 4	cell id	12
	expected cell entry time	1050

Fig. 1. An example UMP.

manages its own UMH which is composed of a limited number of records with the following fields:

- *flag*,
- *UMP identification (UMP Id)*,
- *cell Id*,
  - *expected entry time interval (EETI)*,
  - *the earliest entry time* (t_{ee}),
  - *the latest entry time* (t_{le}),
- *number of visits (NV)*, and
- *next node structures (NN)*
  - *index of the next node* (I_{nc}),
  - *number of times that the next node is visited* (N_{nc}).

Note that the records in a UMH are not related to cells, but to the nodes of a UMP.

When an MT enters a new cell, it starts the procedure shown in Fig. 2 to update its UMH. This procedure is composed of "*if*" statements, and a UMH has generally less than 100 records as explained in Section 5. Therefore, this procedure generally requires a limited number of comparisons (i.e., less than 100), and one or two update operations, which makes it simple enough to run on an MT hardware.

If an MT enters a *new cell*, its UMH probably holds a record for that *new cell*. Therefore, the UMH is searched for the record of the *new cell* first. If a record with the same cell id is found and the *expected entry time* is not larger than the maximum entry interval I_{max}, the N_{nc} field in the current node record are incremented. The previous node is the node from which the MT crosses to the current node.

The expected entry time interval which is the difference between t_{ee} and t_{le} must be lower than I_{max}. The minimum t_{ee} and maximum t_{le} values can be determined as lower and upper control limits by using techniques like statistical quality/process control [26] where the difference between these two values gives I_{max}. If the *cell entry time* is not between t_{ee} and t_{le} when a new cell is entered, the *expected cell entry time interval* for the node is also modified (i.e., t_{ee} is replaced if the cell entry time is lower than t_{ee} or t_{le} is replaced if the cell entry time is higher than t_{le}).

When the maximum value for an NV field is reached, all NV fields are decremented with the exception of the records where the NV value has already become 0. The NV field is used to find the aging UMPs. When the UMH is full and a

```
Procedure UpdateUMH(cell id, entry time,
previous cell id, UMH)
Begin
  1) record←create record(cell id, entry
                              time)
  2) if record exists (record, UMH)
     a) record.NV←record.NV+1
     b) if record.t_ee > entry time
        • record.t_ee←entry time
     c) else if record.t_le < entry time
        • record.t_le←entry time
     d) previous record←get previous record
                                (record, UMH)
     e) increment N_nc(previous record, record)
     f) if N_nc becomes the largest (previous
                              record, record)
        • modify MP Id fields (previous
                                record, UMH)
  3) else
     a) if there is no space(UMH),
        • aging record←find an aging record
                                       (UMH)
        • delete the last node (UMH, aging
                                     record)
     b) insert a new record (record, UMH)
     c) If the previous node exists (previous
                    cell id, entry time, UMH)
        • previous record←find previous
                   record (previous cell id,
                           entry time, UMH)
        • create next node (previous record,
                                     record)
End
```

Fig. 2. The UMH update procedure.

new record is needed to be inserted, one of the records with the lowest NV is replaced with the new record.

When the N_{nc} field pointing to the new cell in the previous record becomes the largest N_{nc} of its record after update, it indicates that the UMP changed starting from the previous node. For example, if we modify *next node* 2 structure of Node 0 as $8-4$ (i.e., $N_{nc}=8$ and $I_{nc}=4$) in Fig. 3, the nodes in the UMP starting from Node 0 become 4 and 5. In this case, we must modify the *UMP Id* fields of Nodes 4 and 5 as 0, and the *UMP Id* fields of Nodes 1, 2, and 3 as 1. Because Node 0 is the member of both UMP 0 and 4, and after this modification, the probability that the MT crosses from Node 0 to Node 4 becomes higher than the probability that it crosses from Node 0 to Node 1.

If there is no record for the node entered in the UMH (i.e., there is no record that has the same *cell id* or the *expected entry time interval* becomes larger than I_{max} after its modification of the cell entry time), the data about the new node is inserted into the first unused record, unless UMH data structure is full. Otherwise, the record with the minimum NV is found, and the *next node* links are followed until the *next node* with an NV greater than the minimum NV or a record with no *next node* data is reached. This ensures that we select the record related to the last node of one of the least used UMPs. The *next node* field in the previous node record is also modified such that it does not point to the node to be replaced anymore. Then, this record

	Flag	UMP Id	Cell Id	**EETI**	**NV**	NN 1	NN 2
0	**U**	**0**	**8**	**1080-** 1140	10	7-1	2-4
1	**UM**	**0**	**12**	**480-** 525	7	7-2	0-0
2	**UM**	**0**	**21**	**510-** 570	7	7-3	0-0
3	**U**	**0**	**12**	**1050-** 1095	7	7-0	0-0
4	**U**	**4**	**12**	**630-** 645	2	2-5	0-0
5	**U**	**4**	**8**	**1020-** 1035	2	0-0	0-0

Fig. 3. An example UMH.

is used to store the data related to the new node. If the previous node is known and there is a record for it in the UMH, the links between the previous and the new node are also created after the data about the new node is inserted into the UMH.

3 Location Update by Using the UMP Scheme

An MT starts a location update process when one of the following happens:

- when the MT is turned on, or returns back to the coverage area of the network, and
- if a new node is entered, which is not a member of the UMP registered in the last location update.

During a registration process, an MT first finds the UMP that it follows. Knowing the current cell and the current time is enough to determine the current UMP. By searching through the UMH, the MT finds the node according to its current cell and time. For example, if the current time is 540 (i.e., 9:00 a.m.) and the current cell id is 12 when an MT is turned on, its current UMP will be UMP 0 for the UMH given in Fig. 3 because 540 is within the expected cell dwell time interval for Cell 12 in UMP 0. At the end of this search, if the MT cannot find a UMP for its current situation, it starts to create a new UMP until it enters an existing UMP. During the time when it cannot register a UMP, a static or a dynamic location update technique [2], [4], [23] may be used. In other words, when an MT does not have a UMP in its UMH for its current location, then any other location update scheme can be used.

After finding its current UMP, the MT creates a UMP registration message, which has the structure illustrated in Fig. 4. The first field of this message is a *header* and consists of two subfields, namely the *flag* and *number of nodes* fields. The *flag* field is a single bit, and the network uses it to interpret the content of the message. If this bit is 0, it indicates that this message is related to a UMP which has been previously registered to the network. In this case, the *number of nodes* field gives the number of nodes in the registration message. The network first modifies that many nodes of the previously registered UMP and uses the modified UMP as the current UMP of the MT. If the *number of nodes* field is 0, it indicates that there is no need to modify the UMP before using it. When the *flag* bit is 1, the meaning

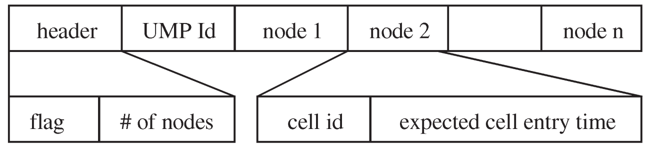

Fig. 4. The structure of a location update message.

of the message depends on the content of the *number of nodes* field. When the *flag* bit is 1 and the *number of nodes* field is greater than 0, either an existing UMP is replaced by a new one or UMP, which was not registered before, is sent to the network. When the *flag* bit is 1 and the *number of nodes* field is 0, it indicates that this registration message is for another location update technique.

The second field in a registration message is the *UMP Identification (UMP Id)* field. It may also be the *location area identification* when the system is in the static location area management mode. The fields for the nodes in the UMP follow the *UMP Id* field. Each of these fields is composed of two subfields, namely the *cell identification* and the *expected cell entry time*. When this structure is created, it is sent to the network in the payload of a registration message.

Note that the network keeps the registered UMPs and modifies them by the registration messages coming from MTs. Therefore, an MT reports only the modifications that are made in the nodes of a UMP since its last registration. It is also possible that the network discards the previously registered UMP of an MT as soon as the MT registers a new UMP. In this case, the MT must send all of the nodes in the UMP as registered to the network for location updates. Hence, when the network keeps only the current UMP and forgets the previously registered UMPs, the average size of registration messages increases. Also, note that our solution is adaptive. If an MT enters a region where it has not been before, it cannot find a UMP in its UMH for the current location. In this case, we can use another location update technique [2], [4], [23].

4 Paging by Using the UMP Scheme

When a call arrives for an MT, the cells registered by the MT in the UMP are paged sequentially starting from the cell where the called MT is most likely to be at the call arrival time. This cell is determined by searching the nodes in the UMP registered by the called MT in order to find the last node which has an expected entry time lower than or equal to the current time, i.e., there is either no node after the last node or the next node has an expected entry time greater than the current time. For example, if a call arrives for the MT which has registered the UMP in Fig. 1, and the current time is 10:30 a.m., Cell 21 is paged first.

If the MT is not found in the first paged cell, the cell with the closest cell residency interval to the current time is paged next. For example, Cell 12 is paged second, and Cell 8 is paged third. One cell at a time search is carried until either a response from the MT is received or a paging delay bound is reached. If the MT is not found in one of the paged cells until the paging delay bound is reached, all remaining cells in the UMP are paged simultaneously. If the MT is still not found, it indicates that the MT either left the coverage

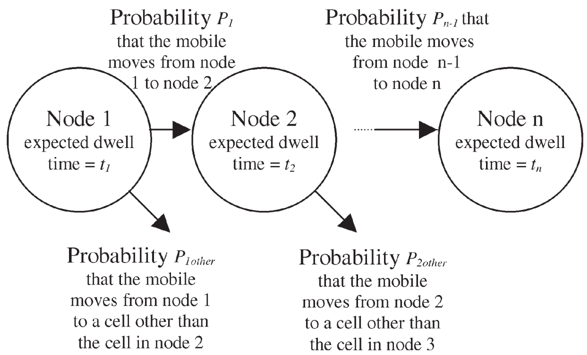

Fig. 5. The model for the user mobility pattern accuracy.

area of the network or turned off without deregistration (i.e., sudden death of power supply) because MTs are supposed to register a new UMP whenever they move out of the registered UMP. In such a case, the called MT is deregistered and recorded as turned off until it registers a UMP. As explained in the location update process in the previous section, whenever the MT returns to the coverage area or is turned on, it starts a location update process.

5 Performance Analysis

In this section, we first provide a model to calculate the *UMP accuracy* γ_i from which we then compare the MT warm-up period and the UMP hit-rate. The *MT warm-up period* is the time required to fill a UMH with UMPs. The UMP hit-rate gives the probability that an MT finds a UMP for its current position in its UMH. The location update and paging costs are also analyzed in this section.

The accuracy of UMP i, γ_i is the ratio of the *successfully completed UMP time* σ_i to the *pattern period* of UMP i, δ_i. The *successfully completed UMP time* σ_i is the time period during which an MT moves according to its registered UMP.

$$\gamma_i = \sigma_i \Big/ \delta_i. \tag{1}$$

The *pattern period* δ_i is the sum of the cell dwell times t_k for all of the nodes k, for $k = 1, \cdots, n$, in UMP i.

$$\delta_i = \sum_{k=1}^{n} t_k. \tag{2}$$

If all cells in UMP i have fixed size, then the pattern period δ_i can be determined from

$$\delta_i = n \Big/ s_i, \tag{3}$$

where s_i is the average speed (cell/hour) of the MT.

In Fig. 5, the nodes in a UMP are represented with circles. As shown in the figure, the successfully completed UMP time δ_i, at which the MT registering UMP i moves according to the registered UMP, is given by

$$\sigma_i = \sum_{k=1}^{n} \left(t_k \prod_{j=1}^{k-1} p_j \right), \tag{4}$$

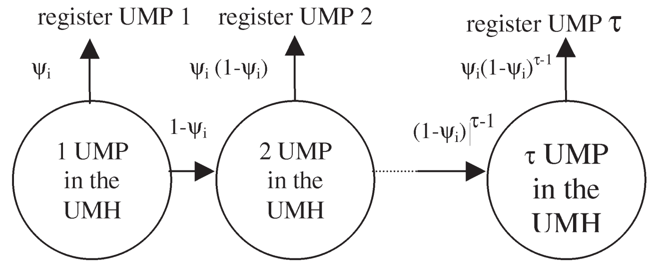

Fig. 6. The model for the user mobility pattern hit ratio.

where p_j is the probability that the MT moves from node j to the next node in UMP i.

For problem tractability and simplicity, we assume that p_j is constant for all nodes in UMP i. The constant p_j is called the *regularity* ψ_i of the MT, which registered UMP i. By using ψ_i, σ_i can be obtained as follows:

$$\sigma_i = \frac{1}{s_i} \sum_{k=1}^{n} \psi_i^{k-1}, \tag{5}$$

From (1), (3), and (5), we determine the accuracy γ_i of UMP i as:

$$\gamma_i = \frac{1 - \psi_i^{s\delta_i}}{s_i \delta_i (1 - \psi_i)}. \tag{6}$$

5.1 The User Mobility Pattern Hit-Rate and the Mobile Terminal Warm-Up Period

In this section, we examine the *UMP hit ratio* and the *MT warm-up period*. The *UMP hit ratio* is the probability of finding a UMP in the UMH of an MT for its current position, and it depends on the number of UMPs in the UMH. The MT warm-up period indicates the time required to enter τ UMPs into a UMH.

The *MT warm-up period* and the *UMP hit-ratio* can be approximated by using the model shown in Fig. 6. The *regularity* ψ_i of MT i is the probability that MT i stays in its most expected UMP. If it moves to a different UMP when there is a single UMP in its UMH, it inserts a new UMP into its UMH. After the insertion of the second UMP, when the MT moves out of its most expected UMP, the probability that it roams according to its second UMP is again equal to its *regularity*. Consequently, the *UMP hit-ratio* increases as the number of UMPs in its UMH increases. Fig. 6 illustrates this relation up to τ UMPs, which is the UMP capacity of a UMH. For the sake of tractability, we assume the capacity of a UMH can be given as the number of UMPs that can be stored, although UMPs may be different in size.

By using the model in Fig. 6, we approximate the *UMP hit-ratio* h_i of MT i when it has τ UMPs in its UMH as follows:

$$h_i = 1 - (1 - \psi_i)^{\tau}. \tag{7}$$

We also use the model in Fig. 6 to approximate the time required to fill a UMH with τ UMPs. If we know the probability of roaming into a new UMP in each location update, we can approximate the number r of location updates expected before roaming into a new UMP by using geometric distribution. However, the probability of having a new UMP decreases every time we insert a new UMP into the UMH as illustrated in Fig. 6. Therefore, the expected number of location updates to reach τ UMPs in a UMH can be approximated by using the geometric distribution with a probability varying every time when a new UMP is inserted into the UMH.

$$r_\tau = \sum_{k=0}^{\tau-1} \frac{1}{(1-\psi_i)^k}, \tag{8}$$

which can be simplified as

$$r_\tau = \frac{1-(1-\psi_i)^\tau}{\psi_i(1-\psi_i)^{\tau-1}}. \tag{9}$$

Using r_τ, we can approximate the required time t_τ to collect τ UMPs by using the average *accuracy* γ_i (6) of UMPs and the average *pattern period* δ_i (3).

$$t_\tau = r_\tau \gamma_i \delta_i. \tag{10}$$

5.2 Location Update Cost

The total location update cost c_r^T for a time interval T is given as follows:

$$c_r^T = \sum_{i=1}^{\eta_r^T} \alpha_i, \tag{11}$$

where

- T is the time interval, for which the location update cost is computed,
- η_r^T is the total number of location updates made during T, and
- α_i is the cost of location update message i.

The total number of location updates η_r^T during the time interval T depends on the *pattern period* δ_i and the *pattern accuracy* γ_i.

$$\eta_r^T = \frac{T}{\delta_i \gamma_i}. \tag{12}$$

The *pattern period* δ_i of UMP i, and the *average speed* s_i of the MT registering UMP i determine the average number of nodes in UMP i. Only the nodes modified since the last location update are notified to the network during a UMP update. Therefore, the number of nodes in a UMP update message is the product of the number of nodes in the UMP and the *irregularity*, i.e., $(1-\psi_i)$.

$$\alpha_i = \delta_i s_i \beta_r (1-\psi_i), \tag{13}$$

where

- s_i is the average speed of the MT,
- β_r is the cost of each node in a UMP update message, and
- ψ_i is the *regularity* of the mobile that is registering UMP i.

5.3 Paging Cost

The total *paging cost* c_p^T during a time interval T can be determined by using (14).

$$c_p^T = \sum_{k=1}^{\eta_p^T} \varepsilon_k \tag{14}$$

where

- n_p^T is the number of the calls received during the time interval T, and
- ϵ_k is the paging cost of call k.

The *number of calls* η_p^T received during the *time interval* T depends on the average *call arrival rate* μ.

$$\eta_p^T = T\mu. \tag{15}$$

The paging cost ε_k for a single call is obtained from

$$\varepsilon_k = \rho_k \beta_p, \tag{16}$$

where

- ρ_k is the number of paged cells, and
- β_p is the cost for each paged cell.

The *number of the cells* ρ_k paged for a call k can be approximated by

$$\rho_k = \begin{cases} \psi_i \sum_{j=1}^{b} j(1-\psi_i)^{j-1} + (\delta_i s_i - b)(1-\psi_i)^b & for\ \delta_i s_i \geq b \\ \psi_i \sum_{j=1}^{\delta_i s_i} j(1-\psi_i)^{\delta_i s_i} & otherwise, \end{cases} \tag{17}$$

where b is the *delay bound* for paging, i.e., the maximum number of cells that can be paged sequentially.

Equation (17) can be expanded as

$$\rho_k = \begin{cases} \frac{1-((1-\psi_i)^b(b+1))+b(1-\psi_i)^{b+1}}{\psi_i} + \\ (\delta_i s_i - b)(1-\psi_i)^b & for\ \delta_i s_i \geq b \\ 1-((1-\psi_i)^{\delta_i s_i}(\delta_i s_i+1))+\delta_i s_i(1-\psi_i)^{\delta_i s_i+1} & otherwise. \end{cases} \tag{18}$$

6 Experimental Results

In this section, we provide the results from the performance evaluation experiments for the metrics explored in Section 5. We also compare the location update and paging performance of the UMP technique against the time-based [4], [15] and movement-based [3], [4], [11] location update techniques, and the blanket [2], [23], selective [16], [22], [25], and velocity [2], [20] paging techniques. In our experiments, 100 MTs are simulated during 50 UMP periods. In each UMP period, new *pattern period*s and average MT speeds are assigned to the MTs randomly. Normal distribution with standard deviation 0.35 is used to generate the random numbers for *pattern period*s. The average *pattern period*s from 4 hours up to 8 hours, and the average MT speeds from 1 cell/hour up to 5 cells/hour are simulated. The simulation times vary based on the completion of 50 UMP periods. Since the *regularity* of an MT has a major impact on the successfully completed part of the registered UMP, the simulation times change according to the given MT regularities. We run simulations for the regularities from 0.5 to 0.99. To evaluate the paging performance, we assume call interarrival times are exponentially distributed.

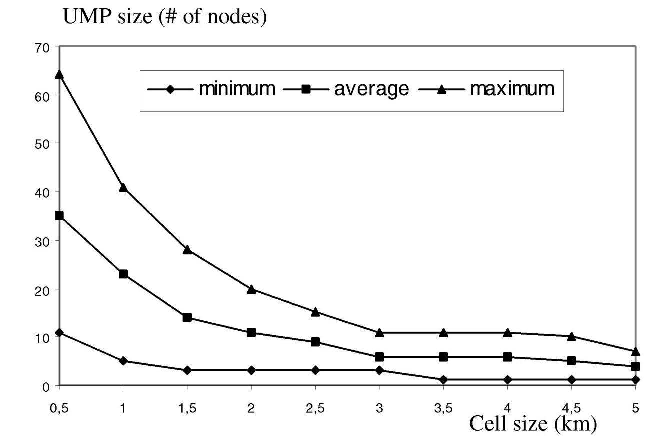

Fig. 7. The average number of nodes in a UMP versus the cell size.

Fig. 8. The regularities of the interviewed people.

6.1 The Average User Mobility History/Pattern Size and Mobile Terminal Regularity

Since, the average number and size (i.e., the number of nodes in a UMP) of UMPs change from person-to-person, city-to-city, country-to-country, and time-to-time, it is not possible to give some generic values for these metrics valid for everywhere. For instance, people live in suburbs and drive to work every morning in some cities, which makes their most used UMP have a large number of nodes. On the other hand, in some cities, people prefer living downtown close to their work, which results in shorter UMPs. Although there are these irregularities related to UMPs, we use the results from a statistical study based on the interviews with 80 people to get an idea about the UMP regularities and sizes. In this study, we observe the locations of the interviewed people for a one-week period. These location data are then converted to a sequence of geographical coordinates. The region where these people live is covered with fixed size hexagons representing cells. Based on this cell layout, the average UMP size and the *regularity* for the interviewed people are determined.

When we apply our scheme to the data obtained by the interviews, we find that the people in our sample space insert on the average six UMPs into their UMH in one week. The rate of creating new UMPs decreases in the following weeks.

In Fig. 7, we give the relation between the average number of nodes in a UMP and the cell size. We show the minimum, average, and maximum number of nodes in a UMP for varying cell sizes. The differences between results are larger for smaller cell sizes. Similarly, the impact of changing the cell size on the number of nodes is larger for the smaller cell sizes. This implies that the UMPs of the people whom we interviewed are confined to relatively small areas and also live close to their work. Therefore, changing the cell size from 4.5 km to 5 km does not make much difference on the number of nodes in a UMP. Based on these findings, the number of nodes in a UMH is expected to be less than 100 for most of the MTs.

In Fig. 8, we illustrate the regularities of the interviewed people. *Regularity* is the probability that an MT moves to the next node from the current node according to its current UMP. We categorize the interviewed people into three groups by observing their regularities. Interestingly, housewives have the lowest *regularity*. They quite often change their UMPs and their UMPs are shorter than the others. Although the average *regularity* of the sales personnel is lower than the average *regularity* of the office workers, the regularities of the sales personnel are less dispersed.

We summarize the results from the *regularity* study as follows:

- All of the people in our sample space have a *regularity* larger than 0.5.
- The regularities of some people are as large as 0.9.
- The average *regularity* of the people in our sample space is 0.72.

6.2 User Mobility Pattern Accuracy

In this section, we examine the *accuracy* γ_i of a UMP given by (6). The accuracies γ_i for the UMP sizes from 1 to 20 nodes are shown in Fig. 9 where each curve represents a *regularity* ψ level. For high *regularity*, the increasing UMP size does not decrease the *accuracy* as much as it does in the lower *regularity* case. The sensitivity of *accuracy* for the changes in the *regularity* level is more pronounced for the higher regularities. For example, the impact of varying *regularity* from 0.9 to 0.99 is much higher than that of varying *regularity* from 0.5 to 0.6.

The *accuracy* for varying *regularity* levels is shown in Fig. 10 where each curve represents a UMP size. Since the MTs move longer, according to their registered UMPs, when their *regularity* is high, the curves get closer to each other as the *regularity* gets higher. This indicates that longer UMPs can be used for *highly regular* users. We also observe that increasing *regularity* improves accuracy even more for higher UMP sizes.

6.3 Mobile Terminal Warm-Up Period and User Mobility Pattern Hit Ratio

In Fig. 11, we show how many location updates are required to collect a given number of UMPs. Note that, each time a mobile needs a location update, it either registers an existing UMP to the network or creates a new UMP if it has not been to the current location before. Since a mobile does not have any data in its UMH at the beginning, it often discovers a new UMP until it enters several UMPs into its UMH. We call this period *MT warm-up period*.

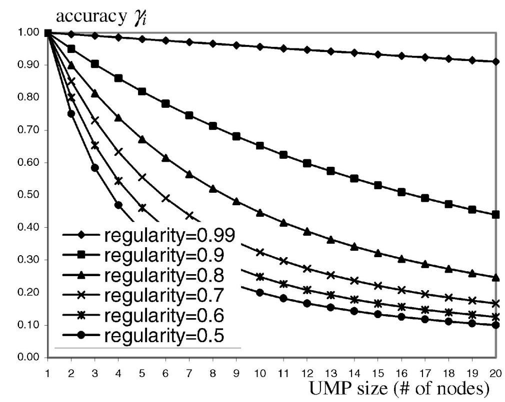

Fig. 9. The accuracy versus the number of nodes in a UMP.

It is intuitively clear that the probability that a highly regular MT discovers a new UMP is lower than that of an irregular MT. This is also observed in Fig. 11. An MT with *regularity* 0.9 needs much more than 500 location updates to insert the fourth UMP into its UMH. However, a mobile with *regularity* 0.7 needs less than 50 updates for the fourth UMP. Fig. 11 also implies that the need for a new UMP decreases exponentially as the number of available UMPs increases.

When a UMP can be derived for the new location from the data existing in a UMH, we call it a *UMP hit*. In Fig. 12, the *UMP hit ratio* is examined for the varying number of UMPs available in a UMH. Increasing the number of available UMPs in a UMH does not make much difference when there are already a number of UMPs in the UMH. According to (7), when a mobile has six UMPs in its UMH, its *UMP hit ratio* becomes more than 95 percent, and its *regularity* does not make much difference.

In Fig. 13, the *UMP hit ratio* for an MT, which has one cell/hour average speed and four hour average *pattern period* is shown starting from the first day of its service activation. Each curve represents a different *regularity* level. In some days, the *UMP hit ratio* of an MT with a lower *regularity* may become higher than that of an MT with a higher *regularity*. Since the lower *regularity* MTs discover new UMPs faster, their *UMP hit ratio* may sometimes become higher for a time interval. However, this does not last long. This is the reason behind their stair-case shape of

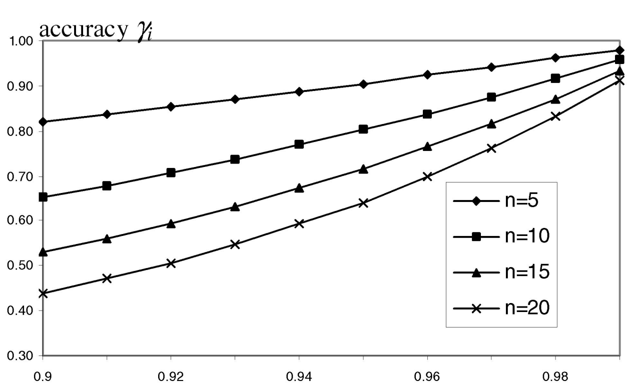

Fig. 10. The accuracy versus the MT regularity.

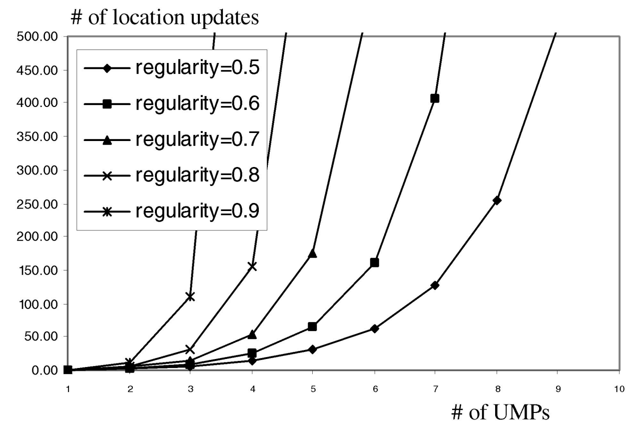

Fig. 11. The number of the required location updates versus the number of the collected UMPs.

the curves. Each time a mobile discovers a new UMP, its *UMP hit ratio* gets higher. The *UMP hit ratio* stays the same until a new UMP is entered into the UMH.

6.4 Location Update and Paging Costs

In Figs. 14 and 15, the *location update rates* (i.e., the number of location updates per hour) are shown for varying *regularity* values. Since the *pattern period* δ_i given by (2) and the average MT speed s_i determine the UMP size, we examine the impact of varying them in the separate graphs. The *pattern period* is fixed to six hours in Fig. 14 where each curve represents a different average MT speed. Modifying the average MT speed has larger impact on the *location update rates* for the lower *regularity* values because the *UMP accuracy* gets higher when the *regularity* gets larger. When the *accuracy* is higher, the probability of staying in a registered UMP is also higher which makes the *location update rate* lower. Since the higher MT speed indicates the larger cell change rate, the *location update rate* is more sensitive to the change in speed in the lower *regularity*.

On the contrary, the *location update rate* is more sensitive to the change in the *pattern periods* in the higher *regularity* because of the same reason. This is shown in Fig. 15. For the

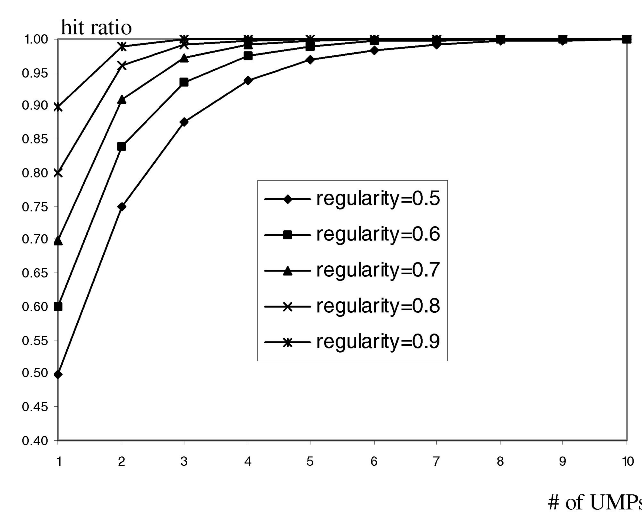

Fig. 12. The hit ratio versus the number of UMPs.

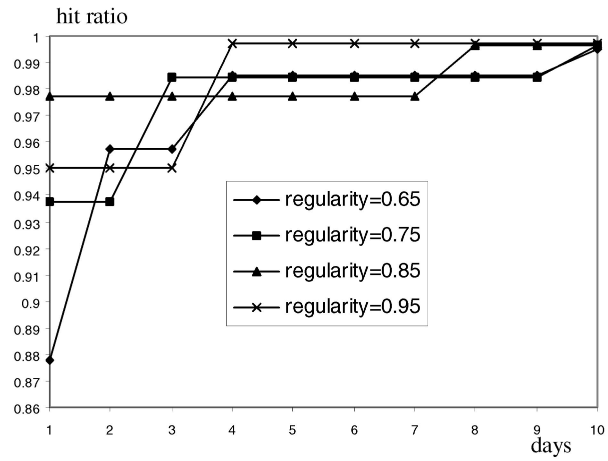

Fig. 13. The hit ratio versus the number of days after an MT starts to operate.

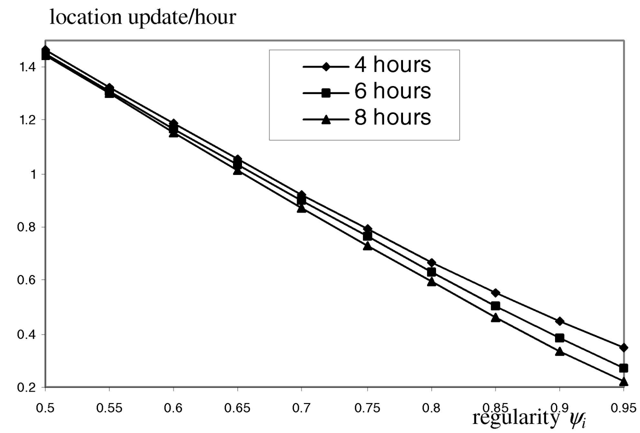

Fig. 15. The location update rate versus the regularity for varying UMP periods.

higher *regularity*, having the longer UMPs makes higher difference in the *location update rate*. When the *regularity* is lower, MTs leave the registered UMPs in the early stages after a location update. Therefore, having a longer *pattern period* does not make much difference.

The average number of the paged cells p_k for each call is examined for varying *regularity* values in Figs. 16 and 17. In Fig. 16, each curve represents a different average MT speed, and, in Fig. 17, each curve represents a different average *pattern period*. The relationship between p_k and the average speed of MT and that the average *pattern period* are similar. When the *regularity* is lower, the probability of not locating the paged MT in the earlier paging becomes higher. Therefore, the impact of modifying the average MT speed or the average *pattern period* is more prominent for the lower *regularity* values.

In Fig. 18, the average number of location updates executed by the simulated MTs are compared for the time-based and UMP schemes. Fig. 18 shows the difference between the number of location updates for the time-based [4], [15] and UMP schemes. Negative values represent the region where the location update cost for the time-based technique is lower. We compare location update cost by changing the *regularity* and the time interval used by the time-based location update technique. In the simulations, we assumed the averaged *pattern period* to be six hours and the average speed is one cell/hour.

When the *regularity* is 0.9, the location update cost of the UMP technique is lower until the time interval for the time-based technique becomes 3.5 hours. When the *regularity* is 0.5, the UMP scheme still performs better until the time interval becomes two hours. For the *regularity* is 0.75, the UMP scheme has lower location update cost compared to the time-based technique. This relationship holds till the time interval for the time-based technique is three hours.

Note that an MT sends only the UMP identification and the data about the nodes modified after the last registration of the UMP during location updates in our scheme. In the real traces used in the experiments in Section 6.1, MTs transfer all nodes in a UMP in 21 percent of location updates. In 57 percent of location updates, only UMP identification is sent. The average number of nodes in a location update message is 2.35. Since our real traces span only one week, the ratio of the location updates, where only UMP identification is transferred, is expected to be higher.

We compare the paging performance of the time-based and UMP schemes in Fig. 19 where the difference in the

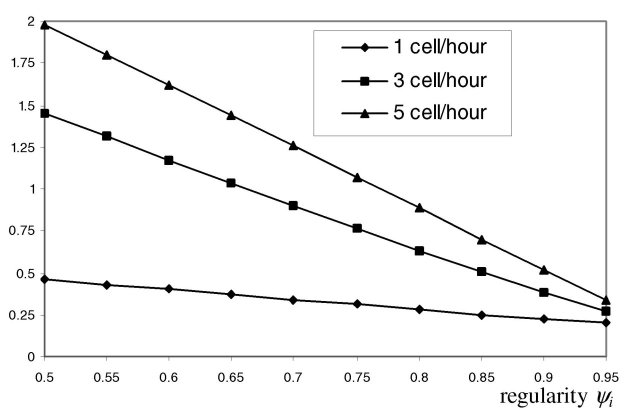

Fig. 14. The location update rate versus the regularity for varying MT speeds.

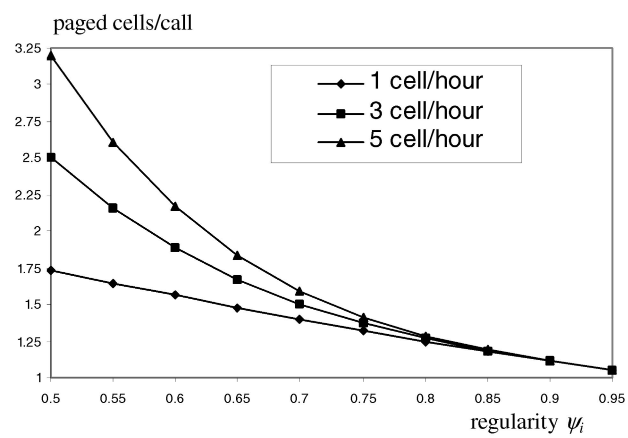

Fig. 16. The paged cells per call versus the regularity for varying MT speeds.

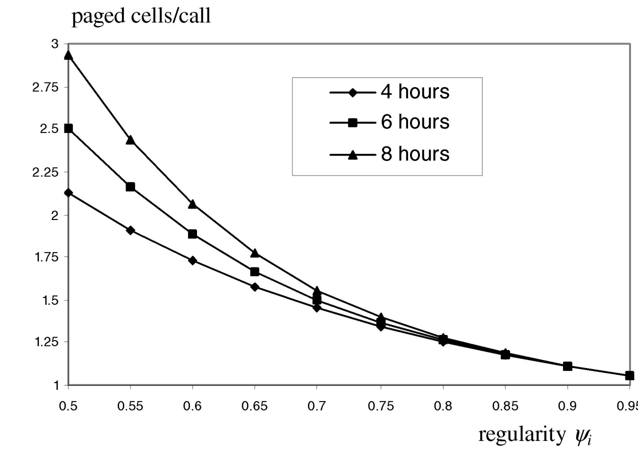

Fig. 17. The paged cells per call versus the regularity for varying UMP periods.

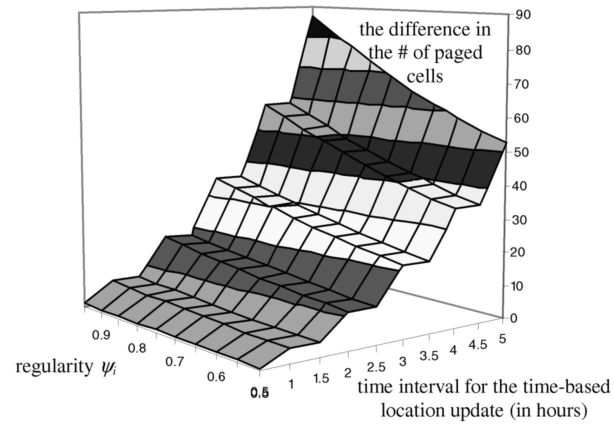

Fig. 19. The comparison of the paging costs between the time-based and UMP techniques.

average number of the paged cells for each call is shown. In these experiments, the blanket polling [23] technique is used with the time-based location update technique [4], [15]. The positive values indicate the region that the UMP scheme decreases the paging cost. In the simulations, we observe that the paging cost of the UMP scheme is always lower, unless the time interval for the time-based technique is so short that a location update is performed for each cell change. The step-like curves are due to the location update time interval and the average MT speed. Since an MT changes a single cell in each hour, a 0.5 hour change in the time interval does not make much difference in the number of cells paged for the time-based technique.

In Figs. 20 and 21, we compare the performance of the UMP scheme with the performance of the movement-based location update technique [3], [4], [11].

For the *regularity* 0.7, the location update traffic created by the UMP scheme is lower than that created by the movement-based technique until the required number of the cell boundary crossings for a new location update is 3. After this point, the movement-based technique decreases the location update traffic. The paging cost of the UMP scheme is always lower than the movement-based technique.

In Figs. 22 and 23, the performance of the selective [16], [22], [23], [25], velocity [2], [20], and UMP-based paging techniques are compared. In the experiments, we used the average speed of the MTs as 1 cell/hour, the average *regularity* of the MTs as 0.7, and the average call interarrival time as 30 minutes. The call interarrival time is exponentially distributed. For the selective paging scheme, LAs are divided into three partitions with location probability 0.7, 0.15, and 0.15 [1]. For the velocity paging scheme [2], [20], the system determines the exact distance that has been covered by the MT since the last registration, and pages only the cells in that distance from the last registered cell.

The reason why the curve for the selective paging is stair-like in Fig. 22 is explained when examining Fig. 19. The UMP scheme outperforms the selective and velocity paging techniques as shown in Fig. 22. For the time interval of one hour, the number of paged cells is 2.4 times more for the selective paging, and 1.3 times more for the velocity paging compared to the UMP scheme. It is 31.5 times more for the selective paging, and 5.4 times more for the velocity paging in the case of five hour time intervals.

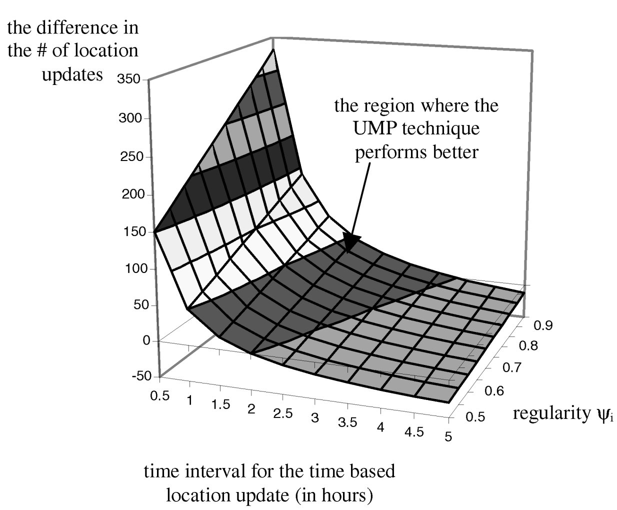

Fig. 18. The comparison of the location update costs between the time-based and UMP techniques.

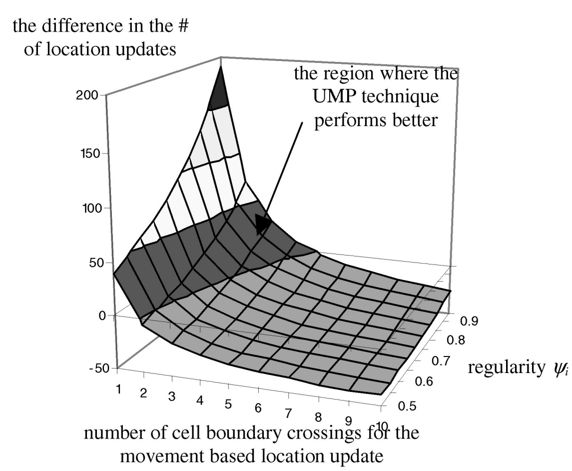

Fig. 20. The comparison of the location update costs between the movement-based and UMP techniques.

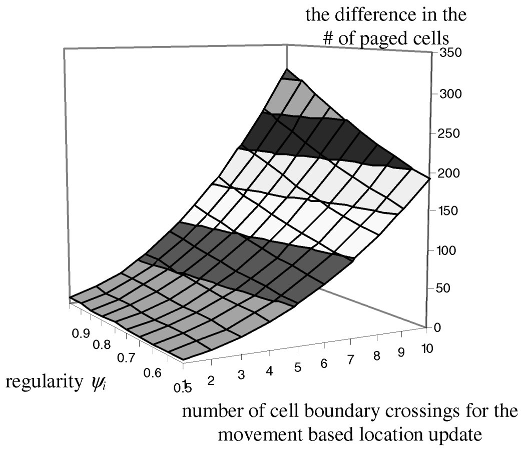

Fig. 21. The comparison of the paging costs between the movement-based and UMP techniques.

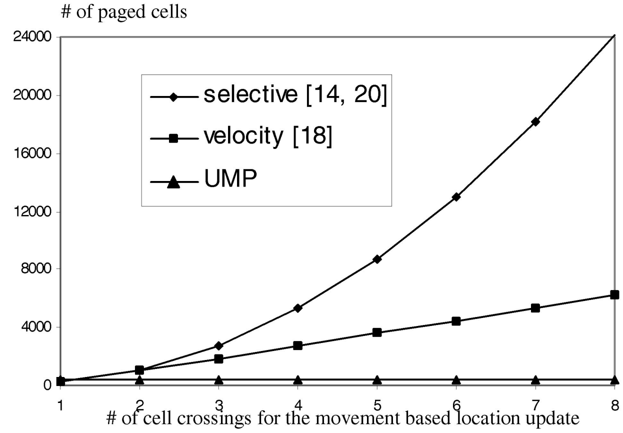

Fig. 23. The paging costs for the selective and velocity paging when the movement-based location update scheme is used.

In Fig. 23, the performance of the UMP, selective and velocity paging techniques are compared for varying movement thresholds, i.e., the number of cell boundary crossings between the movement-based location updates. The UMP technique again clearly outperforms the other techniques.

7 Conclusion

In this paper, we introduce a new location management scheme, namely the user mobility pattern (UMP) scheme, where MTs maintain their history data in a data structure called user mobility history (UMH). During location updates, an MT derives a UMP from its UMH, and registers it to the network. UMPs consist of a number of nodes made up of a cell id and an expected cell entry time. Unless the MT detects that it moves out of the registered UMP, it does not perform another location update. Upon a call arrival for the MT, the cells in its last registered UMP are paged sequentially according to the expected cell entry times. If a delay bound is reached before receiving a reply from the MT, all of the unpaged cells are polled simultaneously.

We carried out analytical, statistical, and simulation-based experiments to evaluate the performance of the proposed scheme. The performance of the UMP scheme is also compared with the time-based and movement-based location update techniques, and the blanket, selective, and velocity paging techniques. The UMP technique creates less location update traffic than the other techniques when reasonable time intervals and movement thresholds are used. It always outperforms the other techniques in paging performance.

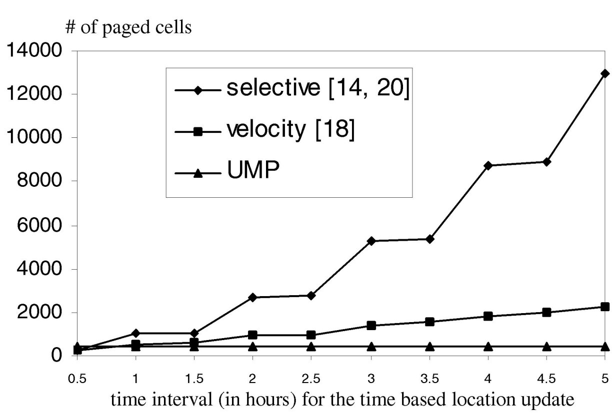

Fig. 22. The paging costs for the selective and velocity paging when the time-based location update scheme is used.

Acknowledgments

This work is supported in part by the US National Science Foundation under grant number CCR-99-88532.

References

[1] A. Abutalep and V.O.K. Li, “Paging Strategy Optimization in Personal Communication System,” *ACM/Baltzer J. Wireless Networks,* vol. 3, pp. 195-204, Aug. 1997.
[2] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu, and W. Wang, “Mobility Management in Next Generation Wireless Systems,” *Proc. IEEE,* vol. 87, no. 8, pp. 1347-1384, Aug. 1999.
[3] I.F. Akyildiz, J.S.M. Ho, and Y-B. Lin, “Movement-Based Location Update and Selective Paging for PCS Network,” *IEEE/ACM Trans. Networking,* vol. 4, pp. 629-638, Aug. 1996.
[4] A. Bar-Noy, I. Kessler, and M. Sidi, “Mobile Users: To Update or not to Update,” *ACM/Baltzer J. Wireless Networks,* vol. 1, no. 2, pp. 175-186, July 1995.
[5] A. Bhattacharya and S.K. Das, “LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks,” *Proc. ACM/IEEE MobiCom ’99,* pp. 1-12, Aug. 1999.
[6] T.X. Brown and S. Mohan, “Mobility Management for Personal Communications Systems,” *IEEE Trans. Vehicular Technology,* vol. 46, no. 2, pp. 269-278, May 1997.
[7] D. Gu and S.S. Rappaport, “A Dynamic Location Tracking Strategy for Mobile Communication Systems,” *Proc. IEEE VTC ’98,* pp. 259-263, May 1998.
[8] J.S.M. Ho and J. Xu, “History-Based Location Tracking for Personal Communications Networks,” *Proc. IEEE VTC ’98,* pp. 244-248, May 1998.
[9] J.S.M. Ho and I.F. Akyildiz, “Mobile User Location Update and Paging under Delay Constraints,” *ACM/Baltzer J. Wireless Networks,* vol. 1, no. 4, pp. 413-425, Dec. 1995.
[10] H.-W. Hwang, M.-F. Chang, and C.-C. Tseng, “A Direction Based Location Update Scheme with a Line-Paging Strategy for PCS Networks,” *IEEE Comm. Letters,* vol. 4, no. 5, pp. 149-151, May 2000.
[11] J. Li, H. Kameda, and K. Li, “Optimal Dynamic Mobility Management for PCS Networks,” *IEEE/ACM Trans. Networking,* vol. 8, pp. 319-327, June 2000.

[12] U. Madhow, M. Honig, and K. Steiglitz, "Optimization of Wireless Resources for Personal Communications Mobility Tracking," *IEEE/ACM Trans. Networking,* vol. 3, pp. 698-707, Dec. 1995.
[13] Z. Noar, "Tracking Mobile Users with Uncertain Parameters," *Proc. ACM/IEEE MobiCom'00,* pp. 110-119, 2000.
[14] C. Rose, "State-Based Paging/Registration: A Greedy Technique," *IEEE Trans. Vehicular Technology,* vol. 48, no. 1, pp. 166-173, Jan. 1999.
[15] C. Rose, "Minimizing the Average Cost of Paging and Registration: A Timer-Based Method," *ACM/Baltzer J. Wireless Networks,* vol. 2, no. 2, pp. 109-116, June 1996.
[16] C. Rose and R. Yates, "Minimizing the Average Cost of Paging under Delay Constraints," *ACM/Baltzer J. Wireless Networks,* vol. 1, no. 2, pp. 211-219, July 1995.
[17] S.K. Sen, A. Bhattacharya, and S.K. Das, "A Selective Location Update Strategy for PCS Users," *ACM/Baltzer J. Wireless Networks,* vol. 5, no. 5, pp. 313-326, Sept. 1999.
[18] J. Sun and H.C. Lee, "Mobility Data Pattern Tracking via Multidimensional Data View Approach," *Proc. IEEE MILCOM'97,* pp. 1255-1260, Nov. 1997.
[19] S. Tabbane, "An Alternative Strategy for Location Tracking," *IEEE J. Selected Areas in Comm. (JSAC),* vol. 13, no. 5, pp. 880-892, June 1995.
[20] G. Wan and E. Lin, "A Dynamic Paging Scheme for Wireless Communication Systems," *Proc. IEEE/ACM MobiCom '97,* pp. 195-203, 1997.
[21] K. Wang, J.-M. Liao, and J.-M. Chen, "Intelligent Location Tracking Strategy in PCS," *IEE Proc. Comm.,* vol. 147, no. 1, pp. 63-68, Feb. 2000.
[22] W. Wang, I.F. Akyildiz, G. Stuber, and B.-Y. Chung, "Effective Paging Schemes with Delay Bounds as QoS Constraints in Wireless Systems," *ACM/Baltzer J. Wireless Networks,* vol. 7, no. 5, pp. 455-466, Sept. 2001.
[23] V.W.-S. Wong and V.C.M. Leung, "Location Management for Next-Generation Personal Communications Networks," *IEEE Network Magazine,* pp. 18-24, Sept./Oct. 2000.
[24] V.W.-S. Wong and V.C.M. Leung, "An Adaptive Distance-Based Location Update Algorithm for Next-Generation PCS Networks," *IEEE J. Selected Areas in Comm. (JSAC),* vol. 19, no. 10, pp. 1942-1952, Oct. 2001.
[25] A. Yener and C. Rose, "Highly Mobile Users and Paging: Optimal Poling Strategies," *IEEE Trans. Vehicular Technology,* vol. 47, no. 4, pp. 1251-1257, Nov. 1998.
[26] http://www.itl.nist.gov/div898/handbook/index.htm, 2002.

Erdal Cayirci (M'97) graduated from the Turkish Army Academy in 1986. He received the MSc degree from Middle East Technical University and the PhD degree from Bogazici University in computer engineering in 1995 and 2000, respectively. He was a visiting researcher with the Broadband and Wireless Networking Laboratory and a visiting lecturer with the School of Electrical and Computer Engineering at the Georgia Institute of Technology in 2001. He is chief of the Combat Models Operations Department at the Turkish War Colleges Wargaming and Simulation Center and a teaching faculty in the Computer Engineering Department at Yeditepe University. His major research interests include sensor networks, mobile communications, tactical communications, and military constructive simulation. He is a member of the IEEE.

Ian F. Akyildiz (M'86-SM'89-F'96) received the BS, MS, and PhD degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively. Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology and director of the Broadband and Wireless Networking Laboratory. His current research interests are in wireless networks, satellite networks, and the next generation Internet. He is an editor-in-chief of *Computer Networks* (Elsevier Science) and an editor for *ACM-Kluwer Journal of Wireless Networks*. He is a past editor for *IEEE/ACM Transactions on Networking* (1996-2001), *Kluwer Journal of Cluster Computing* (1997-2001), *ACM-Springer Journal for Multimedia Systems* (1995-2002), as well as for *IEEE Transactions on Computers* (1992-1996). He guest edited more than 10 special issues for various journals in the last decade. He was the technical program chair of the Ninth IEEE Computer Communications Workshop in 1994, for ACM/IEEE MOBICOM'96 (Mobile Computing and Networking) Conference, IEEE INFOCOM'98 (Computer Networking Conference), as well as IEEE ICC'2003 (International Conference on Communications). He was the general chair of the Eighth ACM MobiCom'2002 Conference. Dr. Akyildiz is an ACM fellow (1996) and a fellow of the IEEE. He received the "Don Federico Santa Maria Medal" for his services to the Universidad de Federico Santa Maria in Chile in 1986. He served as a national lecturer for ACM from 1989 until 1998 and received the ACM Outstanding Distinguished Lecturer Award for 1994. Dr. Akyildiz received the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society) for his paper, entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures," published in the *IEEE Journal of Selected Areas in Communications* (*JSAC*) in January 1996. Dr. Akyildiz received the 2002 IEEE Harry M. Goode Memorial award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking."