韓國經營科學會誌
第33권 第3號
2008年 9月


# 이동 통신망에서 방향성을 지닌 2개의 연속적 위치영역을 이용한 예측 위치 관리 전략

장인갑*† · 홍정식** · 김지표** · 이창훈*

# Predictive Location Management Strategy Using Two Directional Consecutive LAs in a Cellular Network

I.K. Chang* · J.S. Hong** · J.P. Kim** · C.H. Lie*

■ Abstract ■

In this paper, we have presented a dynamic, predictive location update scheme that takes into account each user's mobility patterns. A user's past movement history is used to create two-dimensional transition probability matrix which makes use of two directional consecutive location areas. A mobile terminal utilizes the transition probability to develop a predictive path which consists of several predictive nodes and then the location update is saved as long as a mobile user follows the predictive path. Using continuous-time Markov chain, cost functions of location update and paging are derived and it is shown that the number of predictive nodes can be determined optimally. To evaluate the proposed scheme, simulations are designed and the numerical analysis is carried out. The numerical analysis features user's mobility patterns and regularity, call arrival rates, and cost ratio of location update to paging. Results show that the proposed scheme gives lower total location management cost, compared to the other location update schemes.

Keyword : Predictive Location Update, User Mobility Model, Regularity

## 1. Introduction

Location management is a challenging issue in a cellular network. A cell area is getting smaller to accommodate more mobile users (MUs) with scarce wireless resources. This problem invokes

논문접수일 : 2007년 10월 24일 논문게재확정일 : 2008년 03월 22일
논문수정일(1차 : 2008년 03월 14일)
* 서울대학교 산업공학과
** 서울산업대학교 산업정보시스템공학과
† 교신저자

the need of an efficient location management strategy. Two elemental operations, a location update and paging, play an important role in location management. In the practical cellular network implemented by a location area (LA) based scheme, a location update is the process that an MU reports his new LA information to the network when he changes his current LA. In paging, on the other hand, the network searches for an MU by simultaneously polling all the cells in the last reported LA when an incoming call arrives.

According to literature [4, 5, 11], it is known that the LA-based scheme is not good at reflecting the characteristics of an MU even though it is easy to implement in a cellular network. On the contrary, in cell-based topologies [1, 3, 4, 10], each MU makes up his individual LAs and performs a location update. Such a dynamic scheme may reduce the signaling traffic but it brings about a burden on both networks and user's mobile terminal due to the excessive processing load and power consumption. Therefore, we propose an improved LA-based scheme which exploits individual user's mobile characteristics given the LA information.

In order to reflect individual user's characteristics in LA-based topologies, a profile-based scheme was proposed [5]. The profile-based location scheme utilizes a sequential list of the most likely places each MU can go to for location management. When an incoming call arrives, the network sequentially pages LAs within the list ordered by the steady-state probability. This probability can be computed by the network that maintains the frequency of MUs' visiting each LA. However, this scheme has a problem that paging delay may be too long to make a connection between MUs and the steady-state probability does not reflect the MU's movement direction or pattern well.

To overcome the problem of profile-based scheme and consider individual user's mobility, a selective location update strategy is introduced [11]. When an MU crosses the boundary of an LA, this scheme may not perform a location update on the basis of transition probabilities and cell dwell times. In this scheme, a random walk model is used to develop the user mobility model (UMM) which can reflect MU's movement patterns. However, the problem of determining an optimal update LA set for each MU is very hard to solve because the solution space grows exponentially as the number of LAs increases. Therefore, a genetic algorithm is introduced to obtain approximate solutions. Also, if an MU moves out to non-update LAs, paging delay may be a serious problem because a network searches all candidate LAs until it find the MU.

Considering that most MUs have a routine movement path to a destination and tend to follow the routine path, [4] defines user mobility pattern (UMP). [4, 1, 10] also address the individual mobility patterns to predict dynamically the MU's future location. However, since those schemes should be implemented based on real-time update scheme, networks experience heavy processing loads and a mobile terminal consumes excessive battery power. In addition, requiring redesigning network architectures and maintaining a huge volume of data for each user such as cell IDs, cell entry time, cell residence time, etc., they have a major drawback to implementation in practice.

In this paper, we propose a predictive location management strategy using two directional consecutive LAs. In the proposed scheme, we use the information on two directional consecutive LAs to predict MU's next movement. Based on the two previous consecutive LAs that an MU just passed, the proposed scheme predicts the next LAs to which the MU possibly moves and maintains a list of the predicted LAs. If the MU moves to the predicted LAs, the location update should be skipped. Because the scheme uses two consecutive points (LAs) which indicate the direction of movement, the prediction accuracy of the proposed scheme would be naturally higher than that of other schemes that use only one point. Paging is performed sequentially on the LAs that exist in the list of the predicted LAs. First, the last updated LA is paged. If an MU is not found, the predicted LAs are paged in order they are created. It is shown that the location update cost is reduced dramatically due to the less location updates while small increase in the paging cost is observed.

The remainder of this paper is organized as follows : In section II, we describe the network topology, UMM, and location update and paging algorithm to develop the proposed scheme. Location management cost is derived in section III and simulation is discussed in section VI. Section V addresses the determination of the optimal value and numerical analysis. Finally, we conclude this paper in the last section.

## 2. System Model

Structured graph models for a cellular network have been frequently used in solving the location management problem. Because it is easy to represent either the geometry or the interconnections between cells in the structured graph models, we can simplify the mobility tracking problem and explain it more effectively. Under the LA-based system, the network can be represented by a bound-degree, connected graph $G=(V,E)$, where the node-set $V$ denotes the LAs and the edge-set $E$ denotes the movement paths (i.e. roads, highways etc.) between pairs of LAs. In the proposed scheme, we utilize a directed edge to consider the direction of a movement path. That is, the direction of an edge is determined by the direction of the two consecutive LAs which are part of MU's movement path. In comparison with the cell-based topology, the burden of maintaining movement history profiles in a network or mobile terminal is lightened and the variation of MU's velocity and direction is relatively small because an LA consists of at least more than one cell [6, 7, 9, 11]. Therefore, it is expected that the LA-based scheme is much more appropriate than the cell-based scheme when we consider the routine movement path of an MU and predict MU's location.

The terminologies and notations used in this paper are summarized as follows in <Table 1>.

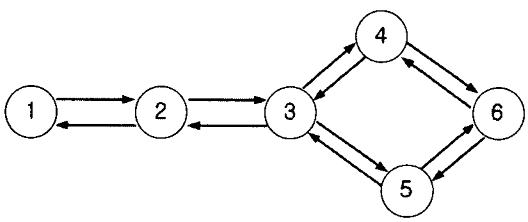
### 2.1 Predictive Scheme Using the Transition Probability Matrix

User mobility model (UMM) plays an important role in developing location management strategies. Generally, an individual MU has inherent mobility patterns and the UMM aims to reflect them. Most studies based on the UMM assume that those patterns include MU's routine movement paths or trips like going to work or school [1, 4, 10]. In this paper, we adopt the UMM to exploit user's mobile characteristics so that the prediction accuracy on user's movement can be improved.

〈Table 1〉 List of terminologies and notations

| Terminology/notation | Description |
|---|---|
| Present node | The node where the last location update occurs |
| Previous node | The node where an MU resides before moving to the present node |
| Predictive node | The node with the highest transition probability from the present node or preceding predictive node |
| $PP_{(i,j)}(n)$ | Predictive path of the previous node $i$ and present node $j$, composed of $n$ consecutive predictive nodes |
| $E_j$ | The set of adjacent nodes of node $j$ |
| $\lambda_c$ | Call arrival rate (CAR) which includes both outgoing calls and incoming calls |
| $\lambda'_c$ | Incoming call arrival rate |
| $P_{i,j}$ | Transition probability from node $i$ to node $j$ |
| $P_{(i,j),(j,k)}$ | Transition probability from node $i$ to node $k$ through node $j$ |
| $\mu_{(i,j)}$ | The rate at which the process leaves state $(i, j)$ |
| $\lambda_{ijk}$ $(=\mu_{(i,\ j)}*P_{(i,j),\ (j,k)})$ | Transition rate from state $(i, j)$ to $(j, k)$ in the edge-based graph model where $k$ is not the predictive node |
| $\lambda_{ijk*}$ $(=\mu_{(i,j)}*P_{(i,j),\ (j,k*)})$ | Transition rate from state $(i, j)$ to $(j, k*)$ in the edge-based graph model where $k*$ is the predictive node |
| $R_{(i,j,s),\ (j,k,s')}$ | State transition rate from $(i, j, s)$ to $(j, k, s')$ in continuous-time Markov chain (CTMC) |
| $\pi_{(i,j,s)}$ | Limiting probability of CTMC |
| $N(i)$ | The number of cells that constitute node $i$ |
| $n$ | The number of predictive nodes that constitute $PP_{(i,j)}(n)$ |
| $U_L$ | Unit cost of location update for an MU |
| $U_P$ | Unit cost of paging for an MU |

Since a random walk model, described by one-dimensional Markov chain, can easily characterize the user traffic flow in PCS networks, it is widely used for the UMM design which is represented by the nodes and the transition probability between the corresponding nodes $i$ and $j$, $P_{(i,j)}$ [11]. The transition probability matrix is obtained from user's past movement history and is utilized to predict MU's next location.




[Figure 1] Example of MU's routine movement paths

For example, there is an MU who has a home and an office at node 1 and 6, respectively, shown in [Figure 1]. Based on the past movement history data collected for 200 days, MU's routine paths from home to office are assorted as follows :

- Movement path *1* : 1-2-3-4-6-4-3-2-1 (100)
- Movement path *2* : 1-2-3-5-6-4-3-2-1 (40)
- Movement path *3* : 1-2-3-5-6-5-3-2-1 (20)
- Movement path *4* : 1-2-3-4-6-5-3-2-1 (40)

Four routine paths from home to office and back to home are identified. The number in the parenthesis is the frequency of movement paths the MU chose to follow. In one-dimensional Markov chain for the random walk model, the transition probability, $P_{(i,j)}$, is estimated by the relative frequencies and given in <Table 2>. For instance, the transition probability from node *2* to node *1*, $P_{(2,1)}$, is 1/2 because the MU arrives at node *2* 400 times but leaves 200 times only for node *1*. Also, if the predictive node $j$ is defined as the next movement node which has the highest transition probability from the present node $i$, the predictive node of node 3, for example, is node 2 because $P_{(3,2)}$ is higher than $P_{(3,4)}$ or $P_{(3,5)}$.

〈Table 2〉 The one-dimensional transition probability matrix(1-TPM)

| State (Node) | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | - | 1 | - | - | - | - |
| 2 | 1/2 | - | 1/2 | - | - | - |
| 3 | - | 1/2 | - | 7/20 | 3/20 | - |
| 4 | - | - | 1/2 | - | - | 1/2 |
| 5 | - | - | 1/2 | - | - | 1/2 |
| 6 | - | - | - | 7/10 | 3/10 | - |

Using the one-dimensional transition probability matrix (1-TPM), we can predict the next movement of an MU from current location. When an MU resides at each node, the predictive node is determined as follows :

- Node *2* : The predictive node is either of node *1* or node *3* because $P_{(2,1)} = P_{(2,3)}$.
- Node *3* : The predictive node is node *2* because $P_{(3,5)} < P_{(3,4)} < P_{(3,2)}$.
- Node *4* : The predictive node is either of node *3* or node *6* because $P_{(4,3)} = P_{(4,6)}$.
- Node *5* : Same as node *4*.

As shown in the example, the model with one-dimensional transition probability sometimes has a problem with determining a predictive node accurately because it does not consider MU's movement direction. For example, assume that an MU goes on the Movement path *1* and the MU is currently at node *3*. Then the MU is presumed to go back to node *2* because node *2* is the predictive node even if the MU just arrived at node *3* through node *2*. Therefore, we could improve the prediction accuracy if we take the direction of a path into account.

〈Table 3〉 The transition probability matrix of the edge-based graph

| State | (1, 2) | (2, 1) | (2, 3) | (3, 2) | (3, 4) | (3, 5) | (4, 3) | (4, 6) | (5, 3) | (5, 6) | (6, 4) | (6, 5) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (1, 2) | - | - | 1 | - | - | - | - | - | - | - | - | - |
| (2, 1) | 1 | - | - | - | - | - | - | - | - | - | - | - |
| (2, 3) | - | - | - | - | 7/10 | 3/10 | - | - | - | - | - | - |
| (3, 2) | - | 1 | - | - | - | - | - | - | - | - | - | - |
| (3, 4) | - | - | - | - | - | - | - | 1 | - | - | - | - |
| (3, 5) | - | - | - | - | - | - | - | - | - | 1 | - | - |
| (4, 3) | - | - | - | 1 | - | - | - | - | - | - | - | - |
| (4, 6) | - | - | - | - | - | - | - | - | - | - | 5/7 | 2/7 |
| (5, 3) | - | - | - | 1 | - | - | - | - | - | - | - | - |
| (5, 6) | - | - | - | - | - | - | - | - | - | - | 2/3 | 1/3 |
| (6, 4) | - | - | - | - | - | - | 1 | - | - | - | - | - |
| (6, 5) | - | - | - | - | - | - | - | - | 1 | - | - | - |

Now, we introduce two-dimensional transition probability matrix (2-TPM) to capitalize on MU's movement direction in which edge *(i, j)* means the path from node *i* to node *j* and $P_{(i,j),(j,k)}$ denotes the transition probability from edge *(i, j)* to edge *(j, k)*. The edge *(i, j)* consists of two consecutive nodes, *i* and *j*, and shows the direction of a path. <Table 3> for the example in [Figure 1] shows 2-TPM, also computed by the relative frequencies of movements. For instance, $P_{(2,3),\ (3,4)}$ = 7/10 because the MU arrives at node *3* from node *2* and leaves for node *4* 140 times out of 200, where nodes *2*, *3*, and *4* mean the previous node, present node, and predictive node, respectively.

If we apply the edge-based transition probability to the example, we can get the predictive node at each present node as follows :

- Node *2* : If an MU moves from node *1* (node *3*), the predictive node will be node *3* (node *1*).
- Node *3* : If an MU moves from node *2* (node *4* or node *5*), the predictive node will be node *4* (node *2*).
- Node *4* : If an MU moves from node *3* (node *6*), the predictive node will be node *6* (node *3*).
- Node *5* : Same as node *4*.

In our particular example, 2-TPM predicts MU's next location more precisely than 1-TPM; in fact, 2-TPM produces all the predictive nodes correctly no matter which movement path the MU takes. That's because 2-TPM considers the direction of an MU with a previous node, which allows eliminating the backward direction among the possible movement directions. Actually, even though there are many routine paths, MU's next movement could be anticipated perfectly with the information from 2-TPM unless MU's routine path shares an edge with other paths.

[7, 8, 12] propose that a mobile terminal can maintain several kinds of data at the same time and transmit them to the network. Even though the size of 2-TPM is double of 1-TPM, the actual amount of data for 2-TMP does not increase in proportion to the size of the matrix. However, it may cause some computing loads on a network which are usually associated with the database handling cost. Hence, it is necessary that we consider a trade-off between the prediction accuracy and the computing loads. In reality, it is difficult to measure the computing loads in practical networks so that we leave the problem for further study.

## 2.2 Location Update and Paging Algorithm

In the existing LA-based scheme, a location update occurs whenever an MU changes his current LA. In the proposed scheme, the location update occurs when the MU deviates from a predictive path. In other words, the location update does not occur as long as the MU moves to a predictive node of the predictive path. The predictive path consists of *n* predictive nodes which should be determined to minimize the location management cost (LMC); the more predictive nodes the predictive path has, the less the update cost but the more the paging cost. The location update is also triggered after an MU departs from the $n^{th}$ predictive node of the predictive path.

When an MU moves to node *j* from node *i* and a location update has occurred, node *j* becomes the present node and the predictive node *k* is determined such that *k* maximizes $P_{(i,j),(j,k)}$ in the proposed UMM using 2-TPM. Similarly, using the information of directional edge $(j, k)$, the next predictive node is determined. This procedure continues until a predictive path, $PP_{(i,j)}(n)$, contains $n$ predictive nodes.

In [Figure 1], suppose that an MU moves from node $1$ to node $2$, a location update has been just made, and $n=3$. Then, nodes $1$ and $2$ become the previous and present node, respectively. Also, $PP_{(1,2)}(3)$ can be set as {3, 4, 6} by using 2-TPM in <Table 3>. Obviously, location update occurs only when the MU deviates from $PP_{(1,2)}(3)$ or departs from the final predictive node, node 6. The proposed location update procedure is explained in [Figure 2 (a)].

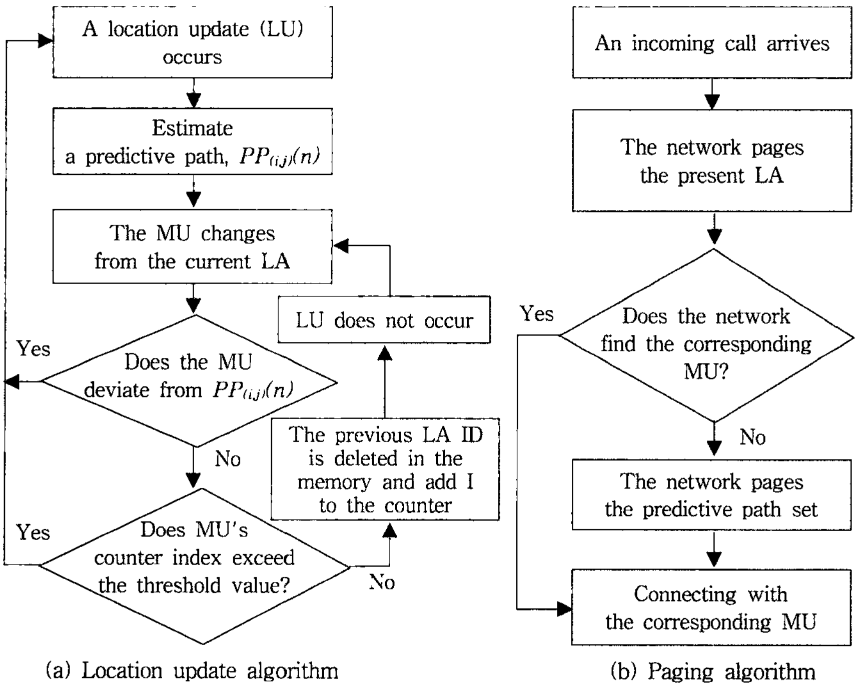
If an incoming call arrives for an MU, sequential paging is performed. The present node is paged first and if the network can not find the MU, all the nodes of $PP_{(i,j)}(n)$ are paged sequentially. Since the unit cost of a location update is much more expensive than that of paging, the proposed scheme can find the optimal $n*$ which minimizes the total location management cost. The paging algorithm is shown in [Figure 2 (b)].

According to the proposed algorithm, it is necessary for a mobile terminal to store $n$ nodes information. Referring to [7, 8, 12], a mobile terminal can maintain several identification lists of nodes and this method can be easily implemented in real cellular networks such as IS-54, IS-95, and GSM.

# 3. Location Management Cost

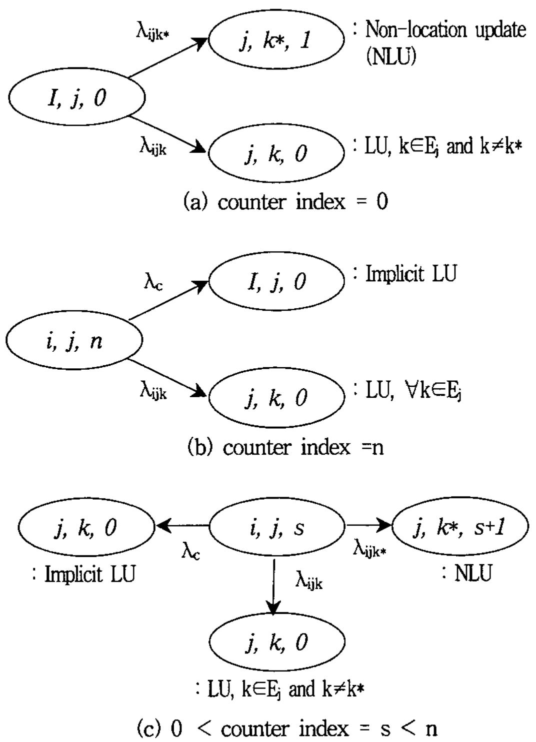
## 3.1 Continuous-Time Markov Chain for the Proposed Scheme

In this section, we employ continuous-time




[Figure 2] The proposed location update and paging algorithms

Markov chain (CTMC) to derive the cost function of location management. CTMC includes two nodes information *(i, j)* and a counter index. The counter index which can have the value from *0* to *n* is added to the state *(i, j)* to check whether the location update should be performed after moving. If the counter index has the value of *0*, it indicates the completion of location update at the current node; otherwise, non-location update occurs because an MU follows the predictive path. We assume that the cell residence time of an MU and inter-arrival time between calls are exponentially distributed with parameter $\pi_{(i,j)}$ and $\lambda_c$. [Figure 3] represents the state transition rates of CTMC in the proposed scheme using two directional consecutive nodes.




[Figure 3] The state transition rates of CTMC in the proposed scheme

In [Figure 3 (a)], the state *(i, j, 0)* means that an MU moves from node *i* to node *j* and a location update is made. That is, node *i* and *j* becomes the previous and present node, respectively. Subsequently, the predictive path $PP_{(i,j)}(n)$ is generated and the counter index set to *0*. If the MU moves to node *k** which is the predictive node, the location update does not occur and the counter index increases by *1*. If the MU moves to any node *k* other than node *k**, the location update should be performed with the counter index = *0*. Then, in this case, the state transition rates can be described as follows :

$$R_{(i,j,0),\ (j,k^*,1)} = \lambda_{ijk^*} = \mu_{(i,j)} \cdot P_{(i,j),\ (j,k^*)} \tag{1}$$

and

$$R_{(i,j,0),\ (j,k,0)} = \lambda_{ijk} = \mu_{(i,j)} \cdot P_{(i,j),\ (j,k)} \tag{2}$$

[Figure 3 (b)] illustrates two states after the counter index reaches *n*. The state transition rates are given by (3) and (4). The equation (3) represents the state that an MU gets a call so that the location update is performed automatically, which is called the implicit location update and excluded from location update cost. The equation (4) means that the MU moves out of the $n^{th}$ predictive node *j* and the location update is performed.

$$R_{(i,j,n),\ (i,j,0)} = \lambda_c = \mu_{(i,j)} \cdot P_c, \tag{3}$$

$$R_{(i,j,n),\ (j,k,0)} = \lambda_{ijk} = \mu_{(i,j)} \cdot P_{(i,j),\ (j,k)} \tag{4}$$

where $P_c$ is a call arrival probability.

[Figure 3 (c)] depicts the states that an MU receives a call or moves to other node when an MU is at the $s^{th}$ predictive node *j*. Obviously, if the MU moves to the predictive node *k**, non-location update occurs and the counter index increases by *1*; otherwise, location update is triggered. The state transition rates are given as follows

$$R_{(i,j,s),(i,j,0)} = \lambda_c = \mu_{(i,j)} \cdot P_c, \quad (5)$$

$$R_{(i,j,s),(j,k,0)} = \lambda_{ijk} = \mu_{(i,j)} \cdot P_{(i,j),(j,k)} \quad (6)$$

$$R_{(i,j,s),(j,k^*,s+1)} = \lambda_{ijk^*} = \mu_{(i,j)} \cdot P_{(i,j),(j,k^*)} \quad (7)$$

Then, the limiting probabilities $\pi_{(i,j,c)}$ are the unique nonnegative solution of the following equations :

$$\mu_{(i,j)} \cdot \pi_{(i,j,s)} = \sum_{m=1}^{n} \sum_{\forall(x,y)} \pi_{(x,y,m)} \cdot R_{(x,y,m),(i,j,s)} + \sum_{\forall(x,y)} \pi_{(x,y,s)} \cdot R_{(x,y,0),(i,j,s)} \ \ for\ s = o, \quad (8)$$

$$\mu_{(i,j)} \cdot \pi_{(i,j,s)} = \sum_{\forall(x,y)} \pi_{(x,y,s-1)} \cdot R_{(x,y,s-1),(i,j,s)} \ \ for\ 1 \le s \le n, \quad (9)$$

and

$$\sum_{\forall(x,y)} \sum_{\forall s} \pi_{(x,y,s)} = 1. \quad (10)$$

### 3.2 Location Update Cost and Paging Cost

In this section, we derive the cost function of the location update and paging. First, the location update cost per unit time for an MU, $C_{LU}$, is given by

$$C_{LU} = \left( \sum_{s=0}^{n-1} \sum_{k,k \ne k^*} \sum_{\forall(i,j)} \pi_{(i,j,s)} \cdot \lambda_{ijk} + \sum_{\forall k} \sum_{\forall(i,j)} \pi_{(i,j,n)} \cdot \lambda_{ijk} \right) \cdot U_L, \quad (11)$$

In (11), the implicit location update is not considered as well as additional data handling costs required to maintain the 2-TPM. The first term of the equation is related to MU's movement which results in a location update where nodes *i*, *j*, and *k* are the previous, present, and non-predictive node, respectively. The second term denotes that the counter index reaches *n* and then a location update is performed regardless of MU's next location.

The paging cost per unit time for an MU, $C_P$, is given by

$$C_P = \left\{ \sum_{\forall(i,j)} \pi_{(i,j,0)} \cdot \lambda'_c \cdot N(j) + \sum_{s=1}^{n} \sum_{\forall(i,j)} \left( \pi_{(i,j,s)} \cdot \lambda'_c \cdot \sum_{v_i \in PP_{(i,j)}(n), r=1}^{s} N(v_r) \right) \right\} \cdot U_P, \quad (12)$$

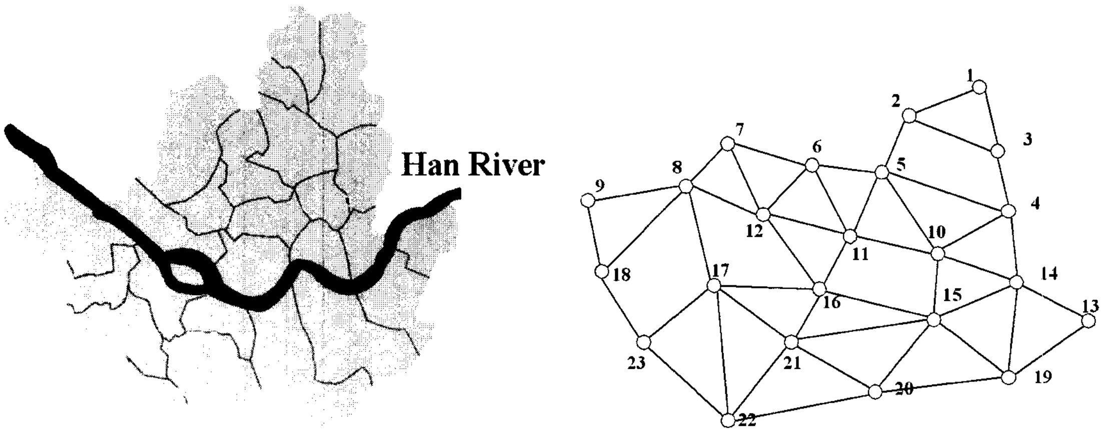
where $v_r$ is the $r^{th}$ predictive node of $PP_{(i,j)}(n)$.

In (12), $\lambda'_c$ is the incoming call arrival rate and assumed to be equal to $\lambda_c/2$ without loss of generality. The first term is about the situation that an MU stays at the present node *j* and is reached immediately by the first paging. The second term means that the sequential paging is performed on the predictive path after an MU moves to a predictive node. By using (12) and (13), the LMC can be obtained as follows :

$$LMC = C_{LU} + C_P. \quad (13)$$

## 4. Simulations

In section II, we explained that the prediction accuracy on MU's next movement can be improved using the information from 2-TPM when an MU has routine paths and in section III, equations to compute the location management cost were derived on the basis of continuous-time Markov chain, where the transition probabilities are also used as major input data. In this section, how adequate data are gathered to build 2-TPM




[Figure 4] Seoul area and its graphical representation for LAs

through simulations is discussed.

In reality, it is hardly possible to get real data on MU's movement paths because of privacy issues. Therefore, sometimes, arbitrary data are used for the convenience [11] but most of relevant studies heavily rely on simulations to obtain data and justify their logical grounds. In our analysis, instead of using arbitrary numbers, a simple simulation procedure is developed to generate data for transition probability between neighboring nodes. An MU is supposed to have one or more routine paths and a certain level of regularity on each routine path.

In literature [4] user's mobile characteristics such as a tendency of following certain route in daily life is defined as user mobility pattern (UMP) and frequency or probability that an MU complies with her UMP is stated as regularity. MUs have different regularities according to their jobs; for instance, office workers may have high regularity while salesperson's regularity seems low. Such an effect of regularity is also considered and tested in our simulation process.

In simulations, we take advantage of geographical information of Seoul in Korea. [Figure 4] shows the Seoul area and its graphical representation for a cellular network which is divided into 23 LAs. In order to simplify the simulation process, it is assumed that the size of every LA is equal, composed of 40 cells. We focus on one specific MU to generate movement history data which will be used to calculate the transition probability. The MU is supposed to have a home LA but many destinations to go. First, we decide the number of UMPs and assign different regularities to them. Each UMP is a routine path which has a predetermined destination from a home LA; other paths have randomly selected destinations. The path between home and destination is determined by the shortest distance method (SDP). Then, the routine paths with regularity are created quite more frequently than other paths. [Figure 5] represents a procedure of simulations.

```
Begin ;
Input parameters {
    G(V, E) information ;
```

```
    Home node ; UMP nodes ;
    Regularity ;
    } ;
Initialization {
    Movement_Profile = Null ;
    } ;
Collect_Movement_Path {
     For(i = 1, i++, Big_Number)
     rn = random_number_generation(0, 1) ;
     Generate_Destination {
    If (rn < regularity,
          A destination is selected among UMPs ;
         Add the UMP to Movement_Profile ;
         Return ;
         Else A destination is selected randomly ;
          The path is determined by SDP ;
         Add the path to Movement_Profile ;
         Return ;
         } ;
      } ;
    } ;
Build_1-TPM{
   Count the frequency (f_ij) between node i to node j
   for all i and j from Movement_Profile ;
   S_i=Sum f_ij for all j ;
   Calculate P_i,j = f_ij /S_i for all i and j ;
    Build 1-TPM ;
    } ;
Build_2-TPM{
   Count the frequency (f_(i,j),(j,k)) between (i, j) to
   (j, k)
   for all i , j, and k from Movement_Profile ;
   S_(i,j) = Sum f_(i,j),(j,k) for all (i, j) ;
   Calculate P_(i,j),(j,k) = f_(i,j),(j,k) /S_(i,j) for all i , j, and k ;
    Build 2-TPM ;
    } ;
End ;
```

[Figure 5] Procedure of simulations

# 5. Numerical Analysis

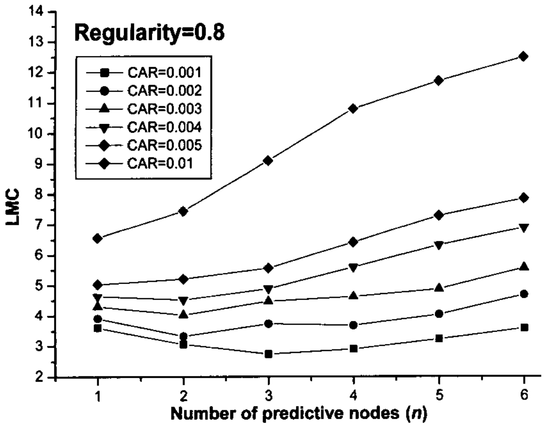
## 5.1 Optimal n*

Numerical analysis is conducted to find out the optimal number of predictive nodes $n*$. The optimal $n*$ that minimizes the location management cost (LMC) does exist because the LMC is the sum of location update cost and paging cost. The cost function of location update is monotonously decreasing as the number of predictive nodes increases because no location update is required as long as an MU moves along the predictive nodes. However, the cost function of paging is monotonously increasing because more predictive nodes mean more paging nodes and eventually, more paging cost. Therefore, the sum of two cost functions should have a point $n*$ that produces minimum LMC, which can be easily shown on a graph, too.

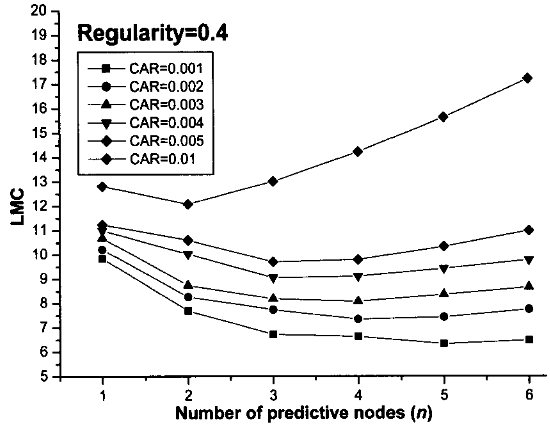
In the next section, optimal $n*$ is sought under various conditions which are mostly related to user's mobile properties like number of UMPs, level of regularity, and incoming call rates, etc. Thus, each MU should have his own optimal $n*$ different from others. Those factors are considered to explain how they affect the optimal $n*$ of a predictive path.

## 5.2 Sensitivity analysis

First, the relationship between number of predictive nodes ($n$) and LMC is analyzed for different call arrival rates (CARs) and regularities when an MU has one UMP and $U_L/U_P$ = 10 . [Figure 6] shows that there exists one specific value $n*$ which minimizes the UMC for each CAR regardless the level of regularity. It is observed that smaller $n*$ is required in order to reduce the paging cost as the CAR increases and high regularity produces lower LMC than low regularity. It is worth to mention that high regularity seems to maintain smaller $n*$ than low regularity in order to balance the location update



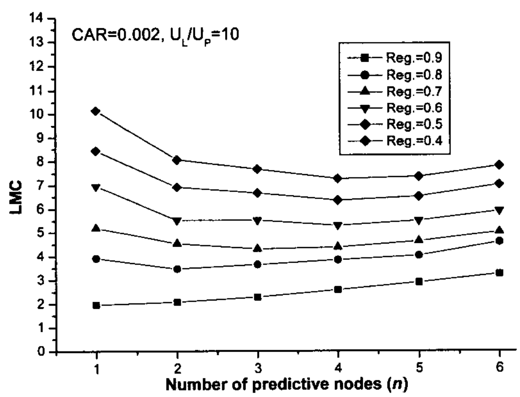

(a) High regularity




(b) Low regularity

[Figure 6] Number of predictive nodes versus LMC for the different CARs

cost and paging cost. In other words, in general an MU with low regularity tends to have bigger $n*$ to cut down the location update cost because low regularity by nature may require more frequent location updates than high regularity. [Figure 7] illustrates the change of the LMC and optimal $n*$ for various levels of regularity in details. For example, as the level of regularity goes up, smaller optimal $n*$ is required because more predictive nodes just incurs unnecessary paging cost.



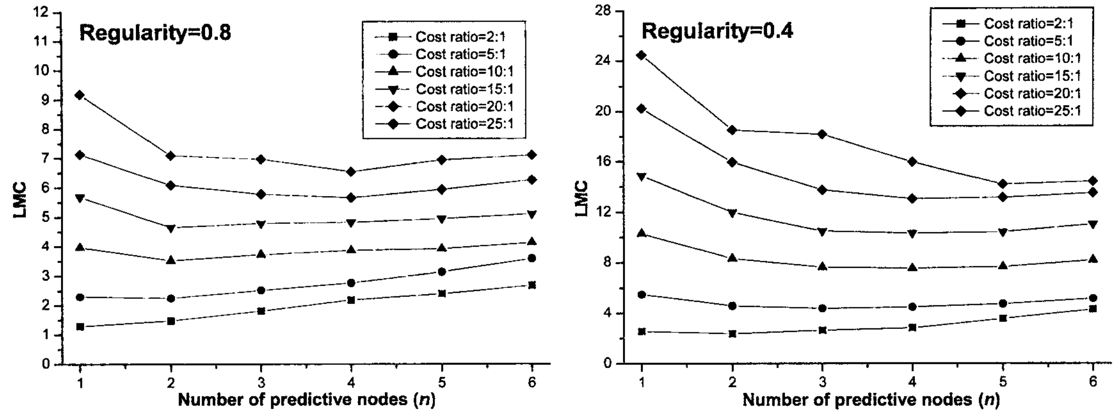

[Figure 7] Number of predictive nodes versus LMC for various levels of regularity

Next, the same investigation into LMC and $n*$ is made for the various ratios of location update cost to paging cost when the CAR is fixed to 0.002. The result is somewhat obvious that it requires bigger $n*$ to contain the location update cost if the cost ratio increases, as shown in [Figure 8].

Finally, the effect of number of UMPs is examined. [Figure 9] depicts that optimal $n*$ and LMC get bigger as more UMPs an MU has ; in [Figure 9 (a)], for example, one UMP corresponds to $n*$ = 2 and LMC = 2.815 while four UMPs to $n*$ = 4 and LMC = 5.602. Therefore, it seems that increase in the number of UMPs turns out the same effect as lowering the level of regularity, which results in more frequent updates. Also, the similar results are observed when more than one UMP are used for the sensitivity analysis.

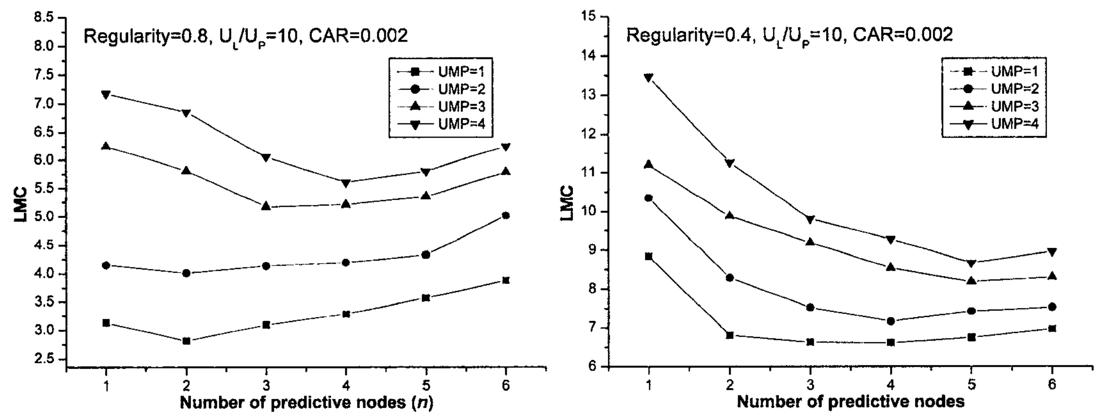
### 5.3 Comparison with other schemes

The proposed scheme is compared with other




(a) High regularity (b) Low regularity

[Figure 8] Number of predictive nodes versus LMC for the different cost ratios

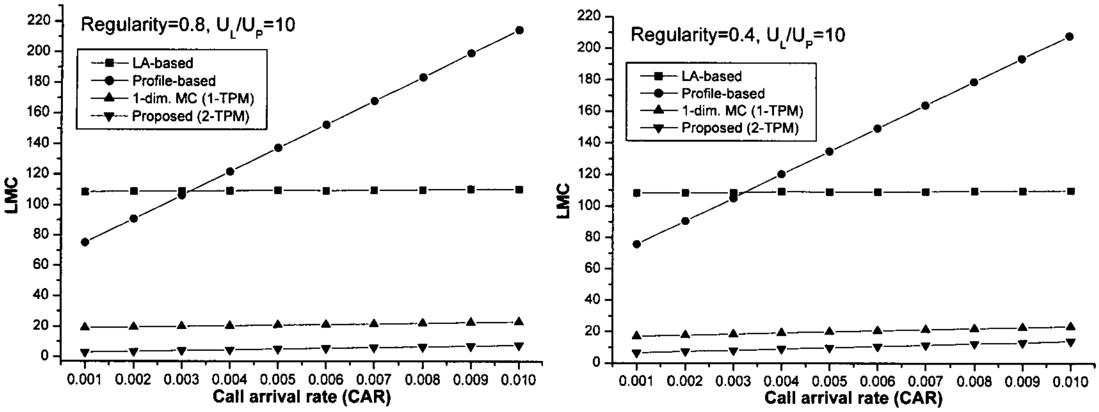



(a) High regularity (b) Low regularity

[Figure 9] Number of predictive nodes versus LMC for the different UMPs

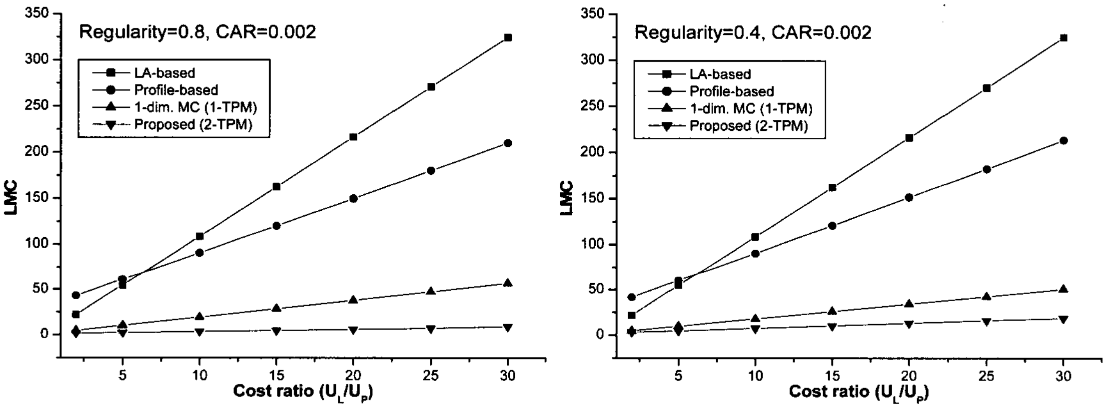
schemes such as LA-based scheme, profile-based scheme, and one-dimensional Markov chain scheme (1-TPM). The LA-based scheme is considered as static because location update is performed independent of each user's mobility and call arrival patterns while the other schemes as dynamic. In the comparisons, based on the result of section 5.2, values 2 and 4 are given to the optimal $n*$ for high and low regularity, respectively.

[Figure 10] shows the relationship between LMC and CAR for various schemes and two different regularities. In the figure, the two schemes, Proposed and 1-dim. MC, outperforms the other two exceedingly because they are more adaptive to user's mobility using the transition probability. By the same reason, the proposed scheme results in less LMC than the 1-dim. MC scheme. It is also noticed that the proposed scheme seems affected much more by the decrease in the level of regularity than the 1-dim. MC scheme because the cost difference between Pro-




(a) High regularity (b) Low regularity

[Figure 10] LMC versus CAR for various schemes




(a) High regularity (b) Low regularity

[Figure 11] LMC versus Cost ratio for various schemes

posed and 1-dim. MC shrinks at low regularity.

[Figure 11] also exhibits that Proposed and 1-dim. MC result in lower cost than the other two for different cost ratios ($U_L/U_P$). Especially, the difference in the LMC between Proposed and 1-dim. MC is enlarged as the cost ratio increases. In other words, the location update cost of the 1-dim. MC scheme grows faster than that of the proposed scheme when the higher location update cost is given, which particularly implies that the proposed scheme anticipates user's next location more accurately than the 1-dim MC scheme so that it requires less location updates.

Overall, the enhanced predictability with 2-TPM makes the proposed scheme more adaptive to user's mobility patterns and more cost-efficient than the compared schemes.

## 6. Conclusions

In this paper, we have presented a dynamic, predictive location update scheme that takes into account each user's mobility patterns. User's past movement history is used to create two dimensional transition probability matrix (2-TPM) which makes use of two directional consecutive nodes. Then, a mobile terminal utilizes the information from 2-TPM to predict user's next movement. The proposed scheme is based on the LA-based topology and relatively easy to implement.

The mobile terminal estimates a predictive path which consists of several predictive nodes (LAs) and then, the location update is saved as long as an MU follows the predictive path. Also, it has been shown that the number of predictive nodes can be determined optimally. Then continuous-time Markov chain is employed to derive cost functions of location update and paging.

To evaluate the proposed scheme, simulations are designed and the numerical analysis is carried out. In the simulations, data are collected to establish 2-TPM. The simulated environment consists of 23 LAs, each LA of 40 cells, representing the geographical area of Seoul. The numerical analysis features user's mobility patterns and regularity, call arrival rates, and cost ratio of location update to paging. Results show that the proposed scheme gives lower total location management cost (LMC), compared to the other location update schemes.

On the other hand, the proposed scheme incurs extra storage and processing requirement at the mobile terminal or base station. Also, the prediction accuracy on user's next movement decreases if the mobile user has two or more routine paths which share at least one edge. However, such extra overhead and prediction deficiency may be outweighed by various merits of the proposed scheme.

## References

[1] Alejandro Quintero, "A User Pattern Learning Strategy for Managing Users' Mobility in UMTS Networks," *IEEE Transactions on Mobile Computing*, Vol.4, No.6(2005), pp.552-566.

[2] Amiya Bhattacharya and Sajal K. Das, "LeZi-Update : An Information-Theoretic Framework for Personal Mobility Tracking in PCS Networks," *Wireless Networks*, Vol.8(2002), pp.121-135.

[3] Amotz Bar-Noy, Ilan Kessler, and Moshe Sidi, "Mobile users : To update or not to update?," *Wireless Networks*, Vol.1(1995), pp. 175-185.

[4] Erdal Cayirci and Ian F. Akyildiz, "User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems," *IEEE Transactions on Mobile Computing*, Vol.1, No.3(2002), pp.236-247.

[5] Gregory p.Pollini and Chih-Lin I, "A Profile-Based Location Strategy and Its Performance," *IEEE Journal on Selected areas in Communications*, Vol.15, No.8(1997), pp.1415-1997.

[6] Ian F. Akyildiz, Joseph S.M. Ho, and Yi-Bing Lin, "Movement-Based Location Update and Selective Paging for PCS Networks," *IEEE/ACM Transactions on Networking*, Vol.4, No.4(1996), pp.629-638.

[7] J.H. Baek and B.H. Ryu, "An Efficient Two-Step Paging Strategy Using Base Station Paging Agents in Mobile Communication Networks," *ETRI Journal*, Vol.26, No.5(2004), pp.493-496.

[8] J.H. Baek, J.Y. Seo, S.K. Lim, and Douglas C. Sicker, "An Enhanced Location-Based Location Update Scheme in Mobile Cellular Networks," *ETRI Journal*, Vol.27, No.4(2005), pp.457-460.

[9] Joseph S.M. Ho and Ian F. Akyildiz, "Mobile user location update and paging under delay constraints," *Wireless Network*, Vol.1(1995), pp.413-425.

[10] Nancy Samaan, and Ahmed Karmouch, "A Mobility Prediction Architecture Based on Contextual Knowledge and Spatial Conceptual Maps," *IEEE Transactions on Mobile Computing*, Vol.4, No.6(2005), pp.537-551.

[11] Sanjoy K. Sen, Amiya Bhattacharya and Sajal K. Das, "A selective location update strategy for PCS users," *Wireless Networks*, Vol.5(1999), pp.313-326.

[12] Y.B. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Transactions on Networking*, Vol.5, No.1(1997), pp. 25-33.