# A Traveling Salesman Mobility Model and Its Location Tracking in PCS Networks*

Ming-Hour Yang†, Lien-Wu Chen†, Yu-Chee Tseng‡, and Jang-Ping Sheu†
†Department of Computer Science and Information Engineering
National Central University, Chung-Li, 32054, Taiwan
‡Department of Computer Science and Information Engineering
National Chiao-Tung University, Hsin-Chu, 30050, Taiwan

## Abstract

*This paper considers the location tracking problem in PCS networks. How a solution to this problem performs in fact highly depends on the mobility patterns of users [14]. In this paper, we propose a new Traveling Salesman Mobility (TSM) model, in hope of catching the mobility patterns of a large group of users. The TSM model is characterized by features of "stop-or-move", "infrequent transition", "memory of roaming direction", and "oblivious in different moves". Then a location tracking strategy based on this TSM model is developed. The scheme only needs to keep very little information for each user. Analyses and simulations are provided, which show that the strategy is very prospective.*

## 1 Introduction

One essential issue in PCS networks is the *location management* or *mobility tracking* problem. To keep its location up-to-date, a mobile subscriber must update its current location with its HLR (home location register) from time to time. On a call arriving, the system will page the subscriber based on its most recent updating. Since it is a tradeoff between updating and paging, considerable research has been done in this topic [1, 2, 3, 4, 5, 6, 8, 10, 11, 12, 13, 15, 17].

The current GSM system adopts the *location area* approach [13]. How to optimally partition LAs is discussed in [16], and the subscribers' moving directions are further taken into consideration in [9]. Dynamic update schemes developed based on users' activity have also been proposed. This can be generally divided into three categories [5]:

1) Time-based: A mobile user registers with its HLR whenever a preset timer expires since its previous update [5, 12].
2) Movement-based: A mobile user registers whenever it has crossed a preset number of cell boundaries since its previous update [2, 5].
3) Distance-based: A mobile user registers whenever the distance between its current cell and its previously registered cell exceeds a preset threshold [3, 5, 8, 11].

*This research is supported in part by the Ministry of Education, ROC, under grant 89-H-FA07-1-4 (Learning Technology) and the National Science Council, ROC, under grants NSC89-2218-E-009-093, NSC89-2218-E-009-094, and NSC89-2218-E-009-095.

On a call arriving, a wireless link between the subscriber and its current base station must be established. The earlier update operation will confine the base stations where the subscriber may be found to a certain range. In reality, paging must be completed within some delay constraint. A simple approach is *single-step paging*, where all cells where the subscriber may reside are paged at once. An alternative is *selective paging* [2, 3, 8], where the cells where the subscriber may reside are partitioned into a number of sets based on the possibility that the subscriber may be located. Then these sets are paged one after another until the called subscriber is found.

How a location tracking strategy performs in fact highly depends on the assumed user *mobility pattern*. As observed by [14], different mobility models, when applied to different schemes, may lead to very different performance conclusions. However, as pointed out in [18], not sufficient studies have been conducted in human's realistic mobility behavior. For example, an office staff will mostly commute between only home and working place. A housewife is more likely to be static than mobile, while a taxi driver may be mobile all the time. Different drivers will have different driving speeds and directions (e.g., an office worker may have quite fixed moving directions, while a taxi driver may have quite random driving directions). Further, when being mobile, a driver may be affected by many situations, such as traffic lights, speed limits of roads, occasional traffic jams, etc. In the literature, most works assumed a simple, but unrealistic, *random walk* model [2, 3, 8]. It is certainly very difficult to have a general model that can catch the mobility patterns of all mobile users. The works by [5, 7] have taken directional bias in user movement into consideration. In [1], the roaming directions of mobile subscribers are considered, and it is assumed that users tend to pick the *shortest paths* leading to their destinations. All these models [1, 2, 3, 5, 7, 8] do not consider the mobility variation of users, since it is unlikely that one will be mobile all the time. In [14], an



1063-6927/01 $10.00 

*activity-based model* is presented, which assumes that users may transit from activity to activity, where an *activity* has an associated time of day, duration, and location. In [6], an efficient way to record a user's roaming history is proposed. While more efficient, the database for [6, 14] could be very large because information is maintained in a per user, per cell/LA basis.

In this paper, we propose a new model called *Traveling Salesman Mobility (TSM)* model. The model is based on the observation from traveling salesmen's daily moving history. We expect that this model will be able to catch the mobility patterns of a large group of users. The model is characterized by the following features: (i) *stop-or-move:* a user is either in a stop or a move state, (ii) *infrequent transition:* the user will transit between stop and move states, but once in a state, the user has a tendency to remain in the same state for quite a while, (iii) *memory of roaming direction:* in a trip, the user will have some preference on some particular direction, and (iv) *oblivious in different trips:* in different trips, the roaming directions have litter correlation. These features will be delineated in more details in Section 2. The first two features are to catch users' mobility variation. The third is to take roaming direction into consideration, while the last is to limit the size of our mobility database.

Based on the TSM model, we then propose a new strategy for location management. The mobility database will be in a per user basis, but not per cell basis. Only recent roaming directions need to be recorded. Our strategy will try to determine users' current states (stop or move). In terms of performance, a subscriber under the stop state will be paged with little cost, while one under the move state will be paged selectively so as to optimize the total cost. Analysis and comparisons are provided, which show that our scheme is very promising.

The rest of this paper is organized as follows. Section 2 discusses our proposed TSM model. Our location management strategy taking account the TSM model is presented in Section 3. In Section 4, we show how to optimize our paging strategy using a selective paging strategy when there is memory in roaming direction. Performance comparisons are presented in Section 5. Conclusions are drawn in Section 6.

## 2 The Traveling Salesman Mobility (TSM) Model

In this section, we propose a new Traveling Salesman Mobility (TSM) model. This model is actually obtained from observing a traveling salesman's daily activity. Undoubtedly, salesmen, especially those traveling constantly, are one of the major user groups of PCS. The model is characterized by the following features.

- **Stop-or-Move:** In the TSM model, we assume that a mobile subscriber will mainly switch between two states: *stop* and *move*. Under the stop state, the subscriber is perhaps working in his/her office, talking to customers, or attending a meeting. From time to time, the subscriber will be mobile and switch to the move state. Under the move state, the subscriber is perhaps in his/her way to home/office or for the next meeting. Once reaching the next destination, the subscriber will enter the stop state. The above scenario may repeat several times in the subscriber's daily life. This can be modeled by a state-transition diagram as in Fig. 1, where the subscriber has probabilities of $p$ and $q$ to remain in the stop and move states, respectively, and probabilities of $1-p$ and $1-q$ to transit to the other state.

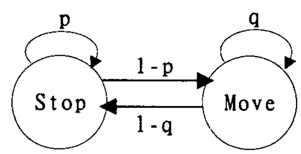


**Figure 1. The state-transition diagram of the TSM mobility model.**

- **Infrequent Transition:** A traveling salesman has a tendency to remain in the same state rather than switching states. That is, if the subscriber is currently in the stop state, it is more likely that the subscriber will remain in the same state in the next moment than switching to the move state. Similarly, once in the move state, the subscriber will remain in the same state until the subscriber arrives at his/her next destination. Reflecting by Fig. 1, we will assume that both probabilities $p$ and $q$ are very close to 1. For instance, one possibility is to set $p = 0.99$ and $q = 0.97$.
- **Memory of Roaming Direction:** Once in the move state, the subscriber's mobility pattern may be affected by many reasons, such as type of vehicles used, traffic jam, and speed limit of different roads. These are certainly very difficult to catch by a general model. However, there should be a destination for this trip. So there will be a tendency in the subscriber's roaming direction. Using a hexagonal cellular system in Fig. 2 as an example, we can use different probabilities to characterize the subscriber's roaming direction to the six neighboring cells. These probabilities should be affected by the subscriber's recent roaming history. But a preference on some directions and a correlation between some roaming directions should exist. For example, directions 0, 1, and 2 may be favored over 3, 4, and 5, and neighboring directions should have similar probabilities.
- **Oblivious in Different Moves:** When the subscriber newly transits to the move state, the subscriber's future roaming pattern should have little relevance to

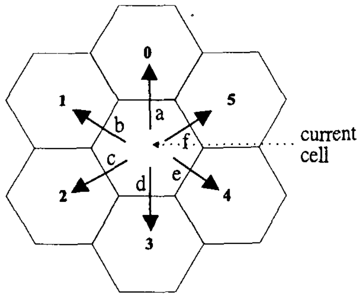


**Figure 2. Roaming directions and their probabilities ($a, b, c, d, e$, and $f$) in a hexagonal system.**

the subscriber's previous roaming pattern. Intuitively, the salesman now has a different destination (and thus roaming direction, for example) from his previous trip. That is, there is memory for the roaming pattern in the *same* trip, but it is "memoryless" between different trips. As a result, the memory of roaming pattern should be refreshed when the subscriber transits from the stop to the move state.

## 3 Update and Paging Strategy

In this section, we present our update and paging strategy. The strategy is developed with an intention to catch the characteristics of the TSM model, and thus optimize the total update and paging cost. Our update strategy will reflect the "stop-or-move" and "infrequent-transition" features of the TSM model. A mobile subscriber will always update, based on its guess, its current state (stop or move) with its HLR. When the subscriber is under the move state, it will use a *movement-based* strategy to update its current location with its HLR. To page the subscriber, we will apply a *selective paging* strategy similar to the work in [2]. On the contrary, when the subscriber is currently under the stop state, we will simply page the cell where the subscriber registered previously. In this case, we will be able to find the user "in one shot."

### 3.1 The Strategy

Since a HLR can only catch a mobile subscriber's mobility at the cellular level, we will interpret the move state in Fig. 1 as a boundary crossing. Similarly, the stop state will also be interpreted as whether the subscriber stays in the same cell or not in the next moment. Based on these interpretations, our strategy needs two constants:

- $D$: the boundary crossing threshold. When a mobile subscriber under the move state makes this number of boundary crossings, it should update with its HLR.
- $T$: the transit-to-stop threshold. When a mobile subscriber under the move state stays in a cell for this number of time units, it should transit to the stop state.

In response to these constants, each subscriber should keep two local variables: (i) $d$, the number of boundary crossings the subscriber has made, and (ii) $t$, the number of time units the subscriber has stayed in the current cell.

When a handset was initially turned on, we assume that it is either in the move or the stop state (this does not affect the correctness of our strategy). The subscriber should update with its HLR based on the following rules.

1. Under the stop state, whenever the subscriber crosses a cell boundary, it should change to the move state and update this fact as well as its current cell with its HLR. Also, the subscriber sets its $d$ to 0 and $t$ to 0.
2. Under the move state, whenever the subscriber experiences a boundary crossing, it should increment its $d$ by 1 and reset its $t$ to 0. Whenever $d$ reaches the threshold $D$, it should update its current location with its HLR again, on which event it should reset its $d$ to 0.
3. Under the move state, the subscriber should increment its $t$ by 1 whenever it stays at the same cell over a duration of one time unit. Whenever $t$ reaches the threshold $T$, the subscriber should change to the stop state and update this fact as well as its current location with its HLR.

When a call arrives, the system will page the subscriber based on the following rules:

1. When the current state of the called subscriber is stop, the system will simply page the cell where the subscriber registered previously.
2. When the current state of the subscriber is move, we will adopt the *selective paging* strategy as in [2] to locate the subscriber. Specifically, we will partition the cells that are at the distance of $D - 1$ from the cell where the subscriber registered previously, into a number of subsets (how to determine these subsets will be discussed in the next section). Then we will page the subset (of cells) with the highest hit probability first. If this fails, the subset with the second highest hit probability will be paged. This is repeated until the subscriber is located.

Note that in the first rule, the system will be able to identify the subscriber "in one shot" because based on our update rules a subscriber under the stop state will always update with its HLR whenever there is a boundary crossing.

### 3.2 Cost Analysis

This section analyzes the total update and paging cost of our strategy under the TSM model. We will apply a Markov model for our analysis. We first define the possible states of a mobile subscriber based on its local variables.

- $S$: The subscriber is under the stop state.

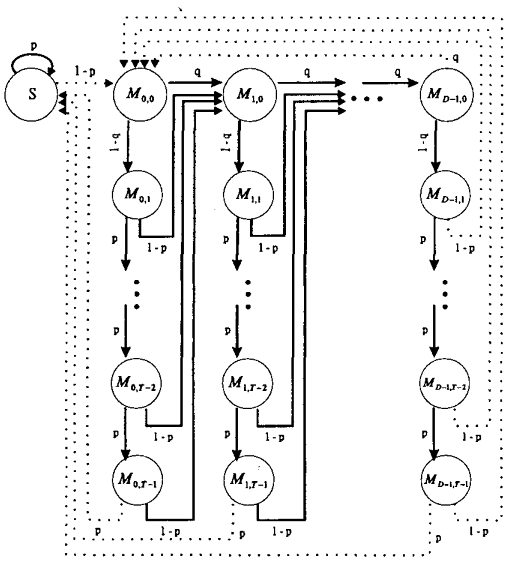


**Figure 3. The state transition diagram of a mobile subscriber under the TSM model.**

- $M_{i,j}, i = 0..D-1, j = 0..T-1$: The subscriber is under the move state, having made $i$ boundary crossings, but having stayed in the current cell for $j$ units of time.

From the above states, we draw a state-transition diagram in Fig. 3. The probability associated with each transition is obtained based on the probabilities in Fig. 1. From this diagram, we need to determine the probability that the subscriber will stay in each state. Let's denote this by $Prob(x)$, where $x$ is any state defined earlier. Since the sum of probabilities over all states must be 1, we have:

$$Prob(S) + \sum_{i=0..D-1, j=0..T-1} Prob(M_{i,j}) = 1.$$

Considering state $S$, from the equilibrium of flows, we have

$$Prob(S)(1-p) = \sum_{i=0}^{D-1} Prob(M_{i,T-1}) \cdot p. \qquad (1)$$

Similarly, we can derive from the equilibrium of flows for state $M_{0,0}$,

$$Prob(M_{0,0}) = Prob(S)(1-p) + Prob(M_{D-1,0}) \cdot q + (1-p)\sum_{j=1}^{T-1} Prob(M_{D-1,j}),$$

and for states $M_{i,0}, i = 1..D-1$,

$$Prob(M_{i,0}) = Prob(M_{i-1,0}) \cdot q + (1-p)\sum_{j=1}^{T-1} Prob(M_{i-1,j}).$$

For the rest of the states, we can derive, for $i = 0..D-1$ and $j = 2..T-1$, that

$$Prob(M_{i,1}) = Prob(M_{i,0}) \cdot (1-q)$$
$$Prob(M_{i,j}) = Prob(M_{i,j-1}) \cdot p\,.$$

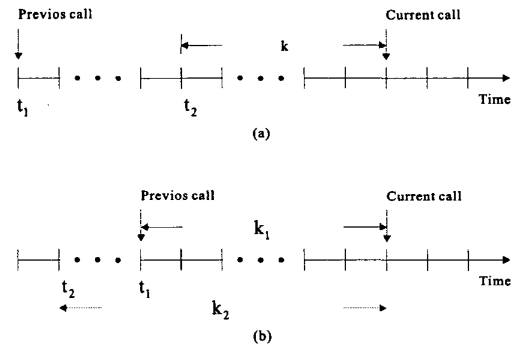


**Figure 4. Relationship of $t_1$ (time of the previous call) and $t_2$ (time of the subscriber entering the move state): (a) $t_1 < t_2$ and (b) $t_1 \geq t_2$.**

There are $DT + 1$ state probabilities to be determined. From the above equations, we can obtain for $i = 0..D-1$ and $j = 1..T-1$ that (note that only those state probabilities that will be used subsequently are shown here)

$$Prob(S) = (1-q)p^{T-1}/(2-p-q)$$
$$Prob(M_{i,T-1}) = \frac{[q+(1-q)(1-p^{T-1})]^i(1-p)(1-q)^2p^{2T-3}}{1-[q+(1-q)(1-p^{T-1})]^D(2-p-q)}$$
$$Prob(M_{D-1,0}) = \frac{[q+(1-q)(1-p^{T-1})]^{D-1}(1-p)(1-q)p^{T-1}}{1-[q+(1-q)(1-p^{T-1})]^D(2-p-q)}$$
$$Prob(M_{D-1,j}) = Prob(M_{D-1,0})(1-q)p^{j-1}$$

Recall that in our strategy there are three events which will trigger a mobile subscriber to update its location: (i) the subscriber switches from stop to move, (ii) the subscriber switches from move to stop, and (iii) under the move state, the subscriber crosses $D$ cell boundaries. These events are illustrated in Fig. 3 by dashes. Let $C_u$ be the cost to perform an undate. Then the average update cost per time unit is:

$$C_{update} = C_u \cdot (Prob(S)(1-p) + p\sum_{i=0..D-1} Prob(M_{i,T-1}) + q\sum_{j=0..T-1} Prob(M_{D-1,j}))$$
$$= C_u \cdot \frac{(1-p)(1-q)p^{T-1}(2-[q+(1-q)(1-p^{T-1}]^D)}{(2-p-q)(1-[q+(1-q)(1-p^{T-1})^D])}.$$

Next, we calculate the paging cost per call. Let the cost to page a cell be $C_p$. Consider the time when a call arrives. There are two possibilities. If the subscriber is under the stop state, then the cost is $C_p$. Multiplying by the probability that the subscriber is under the stop state, the cost is

$$C_{stop} = Prob(S) \cdot C_p.$$

Otherwise, there is a probability of $1 - Prob(S)$ that the subscriber is under the move state. Consider the time $t_1$ when the previous call arrived and the time $t_2$ when the subscriber entered the current move state (refer to Fig. 4). There are two cases.

1. $t_1 < t_2$: If so, there was an update at time $t_2$. The paging cost will depend on the number of boundary crossings (say $k$) that the subscriber has made from $t_2$ to now. Specifically, from $t_2$ to now, the subscriber would update every time when it made $D$ boundary crossings. The probability that the subscriber has made exactly $k$ continuous boundary crossings is $(1-q)q^k$ (i.e., $k$ continuous moves preceded by a stop were made). Also, to satisfy $t_1 < t_2$, there must be no calls arriving from $t_2$ up to now, which has a probability of $e^{-\lambda_c k}$. As a result, the paging cost under the condition that $t_1 < t_2$ is

$$C_{move1} = \sum_{k=0}^{\infty}(1-q)q^k \cdot e^{-\lambda_c k} \cdot PAGE(k \bmod D),$$

where $PAGE(i)$ is the cost to page a subscriber which is under the move state and which has made $i$ boundary crossings before its previous update.

2. $t_1 \geq t_2$: If so, there was an update at time $t_1$. Suppose that there are $k_1$ time intervals from $t_1$ up to now, and $k_2$ time intervals from $t_2$ up to now. Similar to the earlier case, the paging cost will depend on the value of $k_1$, the number of boundary crossing from $t_1$ up to now. The probability that the subscriber has made exactly $k_2$ continuous boundary crossings is $(1-q)q^{k_2}$. The probability that the call prior to the current one happened at $t_1$ is

$$e^{-\lambda_c k_1} * (1 - e^{-\lambda_c}).$$

Since $k_1$ must be less than $k_2$, the paging cost when $t_1 \geq t_2$ is

$$C_{move2} = \sum_{k_2=0}^{\infty}(\sum_{k_1=0}^{k_2-1}(1-q)q^{k_2} \cdot e^{-\lambda_c k_1} * (1 - e^{-\lambda_c}) \cdot PAGE(k_1 \bmod D)).$$

As a result the paging cost per call is

$$C_{page} = C_{stop} + (1 - Prob(S))(C_{move1} + C_{move2}).$$

Summing all the above together, the total update and paging cost of our strategy in one time unit is

$$C_{total} = C_{update} + (\lambda_c) * C_{page} \quad (2)$$

## 4 Cost Optimization with Location Prediction and Selective Paging

One unsolved problem in the previous section is the paging cost $PAGE(i)$, which was defined to be the cost to locate a subscriber which has made $i$ boundary crossings after its previous update (of course, the value of $i$ is unknown to the HLR). If no selective paging is applied, the HLR will search all the cells that are within a distance of $D-1$ from the previous update cell. In this case, $PAGE(i)$ will be independent of $i$, giving

$$PAGE(i) = C_p * (3(D-1)^2 + 3(D-1) + 1).$$

On the contrary, if a selective paging is applied, the above $PAGE(i)$ will change and it is possible to further optimize the paging cost.

In the following, we first show how to predict the subscriber's location under the move state. We will conduct the prediction based on the assumption that the subscriber has a "memory of roaming direction" as discussed in Section 2. Then we will show how to optimize the paging cost by integrating these predictions with $PAGE(i)$.

### 4.1 Location Prediction with Directional Preference

Suppose a subscriber is under the move state. Consider the six roaming directions as 0, 1, 2, 3, 4, 5 in Fig. 2. Based on our "memory of roaming direction" assumption, let's assume that the probabilities that the subscriber will roam from its current cell toward these directions be $a, b, c, d, e, f$, respectively (these probabilities may be obtained from the user's previous roaming pattern under the same move period).

Let $x$ be the cell where the subscriber registered previously. Given any cell $y$, we will derive the probability that the user will be located in cell $y$ after the user made $n$ boundary crossings, denoted as $P_y(n)$. Apparently, $P_x(0) = 1$, which means that without boundary crossing, the user must be located in the original cell.

To resolve our problem, we need a scheme to number cells. Fig. 5 shows our numbering scheme. The numbering is relative to cell $x$.

1. Number the cell on the north of $x$ by 0, and that on the same direction at a distance of $k$ by $0^k$. Similarly, for the other five neighbors of $x$, number them by $1, 2, \ldots, 5$, and those on the same directions at a distance of $k$ by $1^k, 2^k, \ldots, 5^k$. (Refer to the gray cells in Fig. 5(a).)
2. The above numberings (for gray cells) have partitioned the area into 6 sectors of cells. To number the other cells, let's take the cells in the sector bounded by cells $1^i$ and $2^i, i = 1..\infty$, as an example. The cell that will form a parallelogram together with cells $x$, $1^i$, and $2^j$ will be numbered $1^i|2^j$, where "|" means a string concatenation (refer to Fig. 5(b)). The cells in the other sectors are numbered similarly.

Clearly, when $n = 1$, we have

$$P_0(1) = a, P_1(1) = b, P_2(1) = c, \quad P_3(1) = d, P_4(1) = e, P_5(1) = f \quad (3)$$

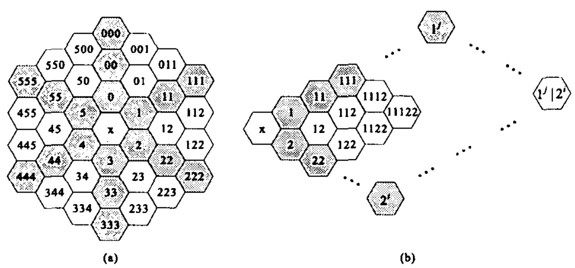

**Figure 5. (a) Numbering of cells with respect to a cell $x$. (b) Numberings based on a parallelogram coordinate.**

For $n > 1$, we will take a recursive approach. Consider any cell $y$. Let $y_0, y_1, \ldots, y_5$ be the six neighbor cells of $y$ along directions 0, 1, ..., 5, respectively. The probability that the user will stay at $y$ after $n$ boundary crossings is the sum of the probabilities that the user stays at the six cells $y_0, y_1, \ldots, y_5$ after $n-1$ boundary crossings, and the last boundary crossing brought the user to $y$. This leads to

$$P_y(n) = P_{y_0}(n-1)\cdot d + P_{y_1}(n-1)\cdot e + P_{y_2}(n-1)\cdot f + P_{y_3}(n-1)\cdot a + P_{y_4}(n-1)\cdot b + P_{y_5}(n-1)\cdot c \quad (4)$$

This equation can be expressed more specifically if the numbering for cell $y$ is known. When $y = 0^i$, we can rewrite Eq. (4) as

$$P_y(n) = P_{y|t(y)}(n-1)\cdot d + P_{y|1}(n-1)\cdot e + P_{r(y)|1}(n-1)\cdot f + P_{r(y)}(n-1)\cdot a + P_{5|r(y)}(n-1)\cdot b + P_{5|y}(n-1)\cdot c,$$

where $t(y)$ is the last element of $y$ (i.e., tail), $r(y)$ is $y$ after removing $t(y)$ (i.e., prefix). In general, for cells $y = i^k$, $i = 0..5$, substituting $a, b, c, d, e, f$ by the probabilities in Eq. (3), we have

$$\begin{aligned} P_y(n) = {} & P_{y|t(y)}(n-1)\cdot P_{[t(y)+3] \bmod 6}(1) \\ & + P_{y|[t(y)+1] \bmod 6}(n-1)\cdot P_{[t(y)+4] \bmod 6}(1) \\ & + P_{r(y)|[t(y)+1] \bmod 6}(n-1)\cdot P_{[t(y)+5] \bmod 6}(1) \\ & + P_{r(y)}(n-1)\cdot P_{t(y)}(1) \\ & + P_{[t(y)+5]|r(y) \bmod 6}(n-1)\cdot P_{[t(y)+1] \bmod 6}(1) \\ & + P_{[t(y)+5]|y \bmod 6}(n-1)\cdot P_{[t(y)+2] \bmod 6}(1). \end{aligned} \quad (5)$$

When $y$ is a mixture of different symbols, we need a different approach. Let cell $y$ be in the sector bounded by the cells $0^i$ and $1^i$, $i = 1..\infty$, we can derive that

$$\begin{aligned} P_y(n) = {} & P_{r(y)}(n-1)\cdot b + P_{y|t(y)}(n-1)\cdot e \\ & + P_{h(y)|y}(n-1)\cdot d + P_{h(y)|r(y)}(n-1)\cdot c \\ & + P_{s(y)}(n-1)\cdot a + P_{s(y)|t(y)}(n-1)\cdot f, \end{aligned}$$

where $h(y)$ is the first element of $y$ (i.e., head), $s(y)$ is $y$ after removing $h(y)$ (i.e., suffix). In general, substituting $a, b, c, d, e, f$ by the probabilities in Eq. (3), we have

$$\begin{aligned} P_y(n) = {} & P_{r(y)}(n-1)\cdot P_{t(y)}(1) \\ & + P_{y|t(y)}(n-1)\cdot P_{[t(y)+3] \bmod 6}(1) \\ & + P_{h(y)|y}(n-1)\cdot P_{[t(y)+2] \bmod 6}(1) \\ & + P_{h(y)|r(y)}(n-1)\cdot P_{[t(y)+1] \bmod 6}(1) \\ & + P_{s(y)}(n-1)\cdot P_{[t(y)+5] \bmod 6}(1) \\ & + P_{s(y)|t(y)}(n-1)\cdot P_{[t(y)+4] \bmod 6}(1). \end{aligned} \quad (6)$$

**Lemma 1** *Let $x = i^m|(i+1 \bmod 6)^{n-m}$, where $i = 0..5$. Then $P_x(n) = C_m^n i^m (i+1 \bmod 6)^{n-m}$.*

## 4.2 Cost Optimization

With the above derivation, we can formulate $PAGE(i)$, which is defined to be the paging cost when the mobile subscriber is known to make $i$ boundary crossings after its previous update. We will adopt a selective paging strategy similar to [2]. Suppose that the previous cell where the subscriber registered is $x$. Based on the selective paging scheme, suppose we divide the cells that are at a distance of $D-1$ from $x$ into $c$ subsets of cells, $S_1, S_2, \ldots, S_c$. Then we will page the subset (of cells) $S_1$ first. If this succeeds, the paging is completed and the cost will be $|S_1|\cdot C_p$. Otherwise, we will page the second subset $S_2$, and the cost will be

$$\left(1 - \sum_{y\in S_1} P_y(i)\right)\cdot |S_2|\cdot C_p,$$

where the leading probability is that for the subscriber not in $S_1$. If this succeeds, the paging is completed; otherwise, $S_3$ will be paged, which will cost

$$\left(1 - \sum_{y\in S_1\cup S_2} P_y(i)\right)\cdot |S_3|\cdot C_p.$$

This will be repeated until last subset $S_c$ is searched. The total cost will be

$$\begin{aligned} PAGE(i) = {} & |S_1|\cdot C_p + \left(1 - \sum_{y\in S_1} P_y(i)\right)\cdot |S_2|\cdot C_p \\ & + \left(1 - \sum_{y\in S_1\cup S_2} P_y(i)\right)\cdot |S_3|\cdot C_p + \cdots + \\ & \left(1 - \sum_{y\in S_1\cup S_2\cup\cdots\cup S_{c-1}} P_y(i)\right)\cdot |S_c|\cdot C_p. \end{aligned}$$

Next, we integrate the above cost into the $C_{total}$ in Eq. (2). This will give the exact cost of our update and

**Table 1. The parameters used for comparison.**

| | |
|---|---|
| boundary crossing threshold ($D$) | $1 \sim 7$ |
| transit-to-stop threshold ($T$) | $1 \sim 7$ |
| paging delay (c) | $1 \sim 3$ |
| call arrival rate ($\lambda_c$) | 0.005, 0.01, 0.1, 1 |
| update-to-paging-cost ratio ($C_u/C_p$) | 1, 10 |
| transition probabilities ($p$, $q$) | $0.7 \sim 0.99$ |
| roaming direction probabilities ($a, b, c, d, e, f$) | (0.6, 0.15, 0.04, 0.02, 0.04, 0.15) |

paging strategy. It remains to determine the way the partition the cells within a distance of $D - 1$ from $x$ into subsets $S_1, S_2, \ldots, S_c$. One simple approach when $D$ and $c$ are small, is to exhaustively test all possible partitions to find the optimal partition which will give the smallest $C_{total}$. We believe that in reality these two values will not be too large.

## 5 Performance Comparisons

In this section, we compare the performances of the movement-based scheme and our scheme based on the above analysis. Under the TSM model, the movement-based scheme will have a total cost of:

$$C'_{total} = \frac{1-p}{(2-q-p)} \cdot \frac{1}{D} \cdot C_u + (3(D-1)^2 + 3(D-1) + 1) \cdot C_p \cdot \lambda_c.$$

(The first term is the probability that a subscriber makes a boundary crossing in one time unit times the probability that this is the $d$-th boundary crossing times the update cost. The second term is the paging cost since we do not use selective paging.) The parameters used in the comparison are in Table 1. In our scheme, an exhausted search is used to find the best partitioning of $S_1, S_2, \ldots, S_c$.

*A) Effects of the Boundary Crossing Threshold $D$*: Fig. 6 shows the costs at various $D$ under different call arrival rates $\lambda_c$. As can be seen, at low $\lambda_c$ (0.005 and 0.01), the movement-based scheme is better when $D$ is small, but will degrade faster than ours as $D$ gradually increases. When $\lambda_c \geq 0.1$, our scheme is better in almost all range of $D$. This is because our scheme pays more update cost to catch the state (move or stop) of the subscribers, hoping in reward of lower paging cost. Thus, at low $\lambda_c$, the benefit will be overwhelmed by the higher update cost. With calls arriving more frequently, the benefit will be more significant. Thus, our scheme is more useful in busy environment.

*B) Effects of Transit-to-Stop Threshold $T$*: In the previous comparison, we used a fixed $T = 3$. Fig. 7 shows the costs at various $T$ under different $\lambda_c$. At lower $\lambda_c$, increasing $T$ will reduce the total cost. On the contrary, at larger $\lambda_c$, increasing $T$ will slightly increase the total cost. This shows an interesting behavior that a larger $T$ will decrease the accuracy in predicting subscribers' states (move or stop). Thus, this may result in a higher paging cost. However, at

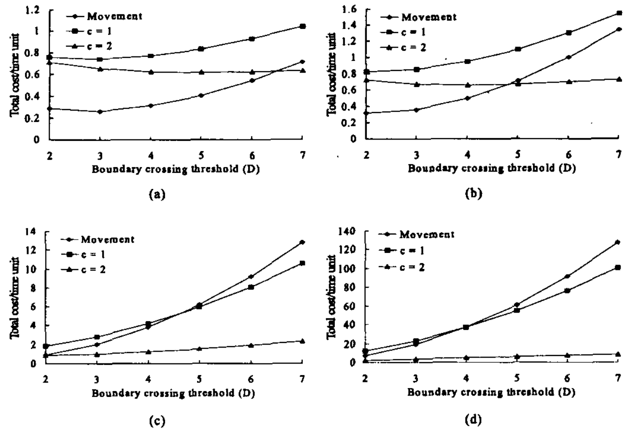


**Figure 6. Costs at various boundary crossing threshold $D$ when (a) $\lambda_c = 0.005$, (b) $\lambda_c = 0.01$, (c) $\lambda_c = 0.1$, and (d) $\lambda_c = 1$ ($T = 3$, $C_u : C_p = 1 : 1, p = q = 0.9$).**

the same time a less number of update messages will be sent. Since the call arrival rate $\lambda_c$ will affect the paging cost, this explains why we see different trends for different $\lambda_c$.

*C) Effects of Call Arrival Rate $\lambda_c$:* Both of the above experiments show that the call arrival rate has some effect on our scheme. To understand this issue, we show Fig. 8 by varying $\lambda_c$. The figure is drawn by separating the update cost and paging cost. As can be seen, the paging cost will increase sharply as $\lambda_c$ increases, while the update cost is quite insensitive to the change of $\lambda_c$. Also, note that we have used $C_u : C_p = 10 : 1$ in this experiment to signify the update cost. Thus, it is worth-while to use our scheme, especially when calls arrive more frequently.

*D) Effect of Transition Probabilities p and q:* Recall that $p$ (resp., $q$) is the probability for a host currently in the stop state (resp., move state) to remain in the same state in the next moment. To understand how these probabilities affect our scheme, we show Fig. 9. The result in Fig. 9(a) shows that a larger $p$ and a smaller $q$ will favor our scheme. The intuition is as follows: (i) a larger $p$ implies a higher probability that a mobile host remaining in the stop state, and thus a larger saving in paging (we may find the host in "one shot"), and (ii) a larger $q$ implies a higher probability that a mobile host remaining in the move state, and thus higher inaccuracy in determining its location when calls arrive. Fig. 9(b) shows the amount of improvement by our scheme as compared to the movement scheme. The range of improvement is about 6 times to 12 times.

## 6 Conclusions

We have proposed a new traveling salesman mobility (TSM) model to characterize mobile subscribers' roaming pattern. Most existing works assumed a random walk roam-