An Optimal Partition Algorithm for Minimization of Paging Costs

Wenye Wang **Ian F. Akyildiz** **Gordon L. Stüber**
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: {wenye,ian,stuber}@ece.gatech.edu

Abstract— **A novel paging scheme under delay bounds is proposed for personal communication systems. This paging scheme is independent of the location probability distributions of the mobile users and satisfies the delay bounds, while minimizing the amount of bandwidth used for locating a mobile user. The proposed paging scheme includes the optimal partition algorithm and paging procedure with respect to paging costs and average delays. The numerical results demonstrate that our proposed scheme is very effective in the minimization of the paging costs for location probability conditions such as uniform and non-uniform distributions.**

Key Words: *Wireless Systems, Paging Area, Partition, Paging Delays, Paging Costs.*

I. INTRODUCTION

Tracking a mobile terminal in wireless systems includes location registration and paging process, both of which cause increasing signaling costs as the demand of wireless services and number of mobile users grow rapidly. Location registration is concerned with reporting the current cell locations. A mobile terminal (MT) registers with the system when it enters a new location area (LA) consisting of a number of cells. Therefore, the system is always aware of the current location of a mobile user. On the other hand, paging is the process in which a system searches for the mobile user by sending poll messages to the cells in the last registered location of the MT.

In particular, paging costs and delay bounds must be considered since the paging cost is associated with bandwidth utilization and delay bounds influence the call setup time in wireless systems. Our objective is to minimize the paging costs under delay bounds. Paging cost is usually measured in terms of cells to be polled before the called user is found [3], [4]. To improve the efficiency of bandwidth utilization, many paging schemes have been proposed, which reduced the paging costs based on location probabilities computed using different methods [5], [8], [9], [13]. Specifically, multi-step paging schemes were suggested to satisfy the delay bounds while reducing the paging costs [1], [2], [7], [10], [11], [12]. In each step, a group of cells called paging area (PA) is searched in one *polling cycle*. A polling cycle is the round trip time from when a paging message is sent until the response is received. The paging delays could then be represented by the required polling cycles; in other words, the number of PAs to be searched before the called MT is found.

On the condition of delay bound, the minimization of paging costs necessitates the partition of an LA into PAs. However, it has been proved that the partition of PAs subject to delay constraint is an NP-complete problem and involves complex and time-consuming computation when an LA is very large [1], [11]. Moreover, most of the previous paging schemes were based on some specific location probability distributions, and they could not provide the optimal partition algorithm to minimize the paging costs.

In this paper, we present an optimal partition algorithm for dividing an LA into PAs, which is also very simple to implement. The rest of this paper is organized as follows. In Section II, we describe the problem formulation in which the paging delay bounds are taken into account. In Section III, the partition algorithm and paging procedure are presented. We demonstrate the performance of the proposed paging scheme and conclude the paper in Section IV and V, respectively.

II. ANALYTICAL MODEL

We assume that each LA consists of the same number of cells, N, in a wireless system. The worst-case paging delay is considered as the delay bound, $\mathcal{D}$, in terms of polling cycle. For instance, if $\mathcal{D}$ is equal to 1, the system should find the called MT in one polling cycle, requiring all cells in the LA to be polled simultaneously. The paging cost, C, which is the number of cells polled to find the called MT, is equal to N. In this case, the average paging delay is at its least value, which is one polling cycle, and the paging cost is at its highest value, $C = N$. On the other hand, when $\mathcal{D}$ is equal to N, the system will poll one cell in each polling cycle and search

This work is supported by NSF under grant NCR-97-04393 and by Korea Telecom – Wireless Communications Research Laboratory (WCRL).

all cells one by one. Thus, the worst-case occurs when the called MT is found in the last polling cycle, which means the paging delay would be at its maximum and equal to N polling cycles. However, the paging cost may be minimized if the cells are searched in decreasing order of location probabilities as demonstrated in [5], [11].

We consider the partition of the LA given that $1 < \mathcal{D} < N$, which requires grouping cells in an LA into smaller PAs under delay bound $\mathcal{D}$. The initial state **P** is defined as **P**= $[p_1, p_2, \cdots, p_j, \cdots, p_N]$, where p_j is the probability of j^{th} cell to be searched in decreasing order of probability. We use triplets $PA^*(i, q_i, n_i)$ to denote the PAs, in which i is the sequence number of the PA; q_i is the probability of the called MT being found in the i^{th} PA, and n_i is the number of cells contained in this PA. Accordingly, the *location probability* q_i of the i^{th} PA is:

$$q_i = \sum_{j \in PA^*(i)} p_j. \tag{1}$$

The *paging cost* under delay bound $\mathcal{D}$, $E[C(\mathcal{D})]$, is computed as follows:

$$E[C(\mathcal{D})] = \sum_{i=1}^{\mathcal{D}} q_i \cdot k_i, \quad \text{where} \quad k_i = \sum_{k=1}^{i} n_k, \tag{2}$$

and the *average delay*, $E[D(\mathcal{D})]$, is

$$E[D(\mathcal{D})] = \sum_{i=1}^{\mathcal{D}} i \cdot q_i. \tag{3}$$

III. Partition Algorithm and Paging Procedure

In order to minimize the paging costs under an upper delay bound $\mathcal{D}$, we develop an optimal partition algorithm and the corresponding paging procedure.

A. Partition Algorithm

The objective of the optimal partition algorithm is to minimize the average number of cells to be searched for the called MT. Given a location probability distribution of a user, there are three necessary conditions to minimize the paging costs.

Lemma If a paging sequence $\mathcal{P}$ satisfies the following conditions of the partition, the *paging cost* $E[C(\mathcal{D})]$ can be minimized:

1. *Probability condition*: The cells must be searched in a decreasing order of the location probability. In other words, if u and v are cells with probabilities $p_u > p_v$, then the optimal paging sequence $PA_{\mathcal{P}}$ that minimizes $E[C(\mathcal{D})]$ must satisfy $u \in PA_{\mathcal{P}}(g, q_g, n_g)$ and $v \in PA_{\mathcal{P}}(h, q_h, n_h)$ for all $g \leq h$.

2. *Forward boundary condition*: It determines the largest probability cell (i.e., with the largest location probability) in a PA. We denote that the MT can be found in the i^{th} PA with probability q_i, as defined in (1), and p^1_{i+1} is the largest probability cell in the $(i+1)^{th}$ paging area with n_{i+1} cells. Then, $p^1_{i+1} \cdot (n_{i+1} - 1)$ must be less than or equal to q_i. This condition implies that the largest probability cell in the $(i+1)^{th}$ PA can not be moved "forward" to the i^{th} PA that is prior to the $(i+1)^{th}$ PA.

3. *Backward boundary condition*: It chooses the smallest probability cell (i.e., with the smallest location probability) in the PA. The backward boundary condition demands that q_i should be less than or equal to $p^s_i \cdot (n_{i+1} + 1)$, where p^s_i is the smallest probability in the i^{th} paging area, and n_{i+1} is the number of cells in the $(i+1)^{th}$ PA. Thus, the smallest probability cell p^s_i can not be moved "backward" to the $(i+1)^{th}$ PA, which comes after the i^{th} PA.

Proof With regard to the probability condition, suppose the paging scheme $\mathcal{P}$ is optimal, but there exists $u \in PA_{\mathcal{P}}(g, q_g, n_g)$ and $v \in PA_{\mathcal{P}}(h, q_h, n_h)$ with $p_u > p_v$ for $g > h$. Let $\mathcal{P}'$ denote the new paging sequence derived from $\mathcal{P}$. In this new sequence, u and v are swapped so that $u \in PA_{\mathcal{P}'}(g, q'_g, n_g)$ and $v \in PA_{\mathcal{P}'}(h, q'_h, n_h)$. We note that

$$\begin{aligned} & E[C(\mathcal{D})] - E[C'(\mathcal{D})] \qquad (4) \\ = \; & \Big(\sum_{i=1,\, i \neq g,h} k_i \cdot q_i + k_g \cdot q_g + k_h \cdot q_h \Big) \\ & - \Big(\sum_{i=1,\, i \neq g,h} k_i \cdot q_i + k_h \cdot q_g + k_g \cdot q_h \Big) \\ = \; & k_g \cdot p_u + k_h \cdot p_v - k_g \cdot p_v - k_h \cdot p_u \\ = \; & (k_g - k_h)(p_u - p_v) > 0, \end{aligned}$$

where k_g is larger than k_h for $g > h$ according to the definition in (2). This is a contradiction to the assumed optimality of $\mathcal{P}$. Therefore, the probability condition in the Lemma is necessary.

Given that the first probability condition is satisfied, the cells can be organized in a non-increasing order of probabilities. Let p^s_i be the smallest probability in the i^{th} PA and p^1_{i+1} be the largest probability in the $(i+1)^{th}$ PA. The *paging cost* C_1 of an MT being found in the $(i+1)^{th}$ PA according to partition $\mathcal{P}$ is calculated from:

$$C_1 = \bar{n}_{i-1} + q_i \cdot (k_{i-1} + n_i) + q_{i+1} \cdot (k_{i-1} + n_i + n_{i+1}), \tag{5}$$

where $\bar{n}_{i-1} = \sum_{l=1}^{i-1} q_l \cdot k_l$, and k_l is defined in (2). If we move the largest probability cell p^1_{i+1} in the $(i+1)^{th}$ partition to the i^{th} partition, the *paging cost* C_2 is then determined by

$$\begin{aligned} C_2 \; = \; & \bar{n}_{i-1} + (q_i + p^1_{i+1}) \cdot (k_{i-1} + n_i + 1) \\ & + (q_{i+1} - p^1_{i+1}) \cdot (k_{i-1} + n_i + n_{i+1}). \end{aligned} \tag{6}$$

Since we do not want to lose the optimality of the partition, the following condition must be satisfied: $C_1 \leq C_2$, which produces

$$\begin{aligned} & q_i \cdot (k_{i-1} + n_i) + q_{i+1} \cdot (k_{i-1} + n_i + n_{i+1}) \\ & \quad \leq (q_i + p^1_{i+1}) \cdot (k_{i-1} + n_i + 1) \\ & + (q_{i+1} - p^1_{i+1}) \cdot (k_{i-1} + n_i + n_{i+1} - 1). \end{aligned} \tag{7}$$

After the simplification, we obtain the following result:

$$p_{i+1}^1 \cdot (n_{i+1} - 1) \leq q_i. \tag{8}$$

This is exactly the forward condition in the Lemma.

In a similar way, we can move the smallest probability cell p_i^s in the i^{th} partition backward to the $(i+1)^{th}$ partition. Accordingly, the *paging cost* C_3 is determined by

$$\begin{aligned} C_3 &= \bar{n}_{i-1} + (q_i - p_i^s) \cdot (k_{i-1} + n_i - 1) \\ &\quad + (q_{i+1} + p_i^s) \cdot (k_{i-1} + n_i + n_{i+1}). \end{aligned} \tag{9}$$

Due to the requirement of the optimality, we have $C_1 \leq C_3$. Consequently, we obtain the following formula

$$q_i \leq p_i^s \cdot (n_{i+1} + 1), \tag{10}$$

which is the backward condition in the Lemma.

B. Paging Procedure

In this section, we illustrate the paging procedure in which the paging areas are constructed in accordance with the optimal partition algorithm in the previous part. We first partition the LA into a series of PAs in such a way that all PAs consist of approximately the same number of cells, followed by testing the boundary conditions described in the previous section.

- **Step 1:** Calculate the number of cells in each PA as

$$n_0 = \left\lfloor \frac{N}{\mathcal{D}} \right\rfloor, \tag{11}$$

and determine the variable k as: $k = N - n_0\mathcal{D}$.

- **Step 2:** Determine a series of PAs as $PA^0(1)$, $PA^0(2)$, $\cdots$, $PA^0(\mathcal{D})$ with the location probabilities of $q_1, q_2, \cdots, q_{\mathcal{D}}$, respectively. n_0 cells are allocated to each of the first $(\mathcal{D} - k)$ PAs, and $(n_0 + 1)$ cells are assigned to each of the remaining k PAs. For example, the first PA consists of n_0 cells and the last PA, i.e., $\mathcal{D}^{th}$ PA, consists of $(n_0 + 1)$ cells.
- **Step 3:** Test the first PA using the backward boundary condition in Section III-A. If $q_1 > p_1^s \cdot (n_2 + 1)$, then p_1^s will be moved to the second PA. Otherwise, keep the partitions obtained from Step 2. We keep testing the first PA until the backward condition is satisfied.
- **Step 4:** Test the first PA using the forward boundary condition in Section III-A. If $q_1 < p_2^1 \cdot (n_2 - 1)$, then p_2^1 will be moved to the first PA. If this movement occurs, we go back to Step 3 in which the backward boundary condition will be tested again. This procedure continues iteratively until the forward condition is satisfied, that is, $q_1 \geq p_2^1 \cdot (n_2 - 1)$.
- **Step 5:** Test the second PA using forward and backward boundary condition as in Step 3 and 4. This procedure continues until each PA has been tested and meets the conditions described in the Lemma. The finalized partitions will be the optimal paging sequence which produces the minimum paging costs.
- **Step 6:** The system polls n_1 cells in $PA^*(1, q_1, n_1)$ first, followed by searching $PA^*(2, q_2, n_2)$, and so forth. The paging procedure stops when the called MT is found.

This paging procedure guarantees that the conditions described in the partition algorithm can be satisfied. Thus, the paging costs are minimized under delay bounds.

Example: Suppose there are 10 cells in an LA, and the upper delay bound is assumed to be 4 polling cycles. Also assume that the probabilities of an MT being found in each cell in the LAs are as follows: 0.35, 0.15, 0.15, 0.10, 0.05, 0.05, 0.05, 0.04, 0.03, and 0.03.

According to Step 1 and 2, we first calculate $n_0 = \lfloor 10/4 \rfloor = 2$ and $k = (N - n_0 \cdot \mathcal{D}) = 2$. Then each of the first $(\mathcal{D} - k) = 2$ PAs consists of $n_0 = 2$ cells, and each of the remaining $k = 2$ PAs consists of $(n_0 + 1) = 3$ cells. This results in the following paging sequence: $PA^0(1, 0.5, 2)$, $PA^0(2, 0.25, 2)$, $PA^0(3, 0.15, 3)$ and $PA^0(4, 0.1, 3)$, as shown in Fig. 1. Under Step 3, we test the first $PA^0(1, 0.5, 2)$ using backward boundary condition. Here, $q_1 = 0.5$, $p_1^s = 0.15$, $n_2 = 2$; therefore, $p_1^s \cdot (n_2 + 1) = 0.45 < q_1$. This means the backward boundary condition is not satisfied, and the cell p_1^s must be moved backward to the second PA, as illustrated between the last two rows in Fig. 1. The new paging sequence becomes $PA^1(1, 0.35, 1)$, $PA^1(2, 0.40, 3)$, $PA^1(3, 0.15, 3)$ and $PA^1(4, 0.1, 3)$. We then apply backward boundary condition again in which $q_1 = 0.35$, $p_1^s = 0.35$, $n_2 = 3$. It is observed that $p_1^s \cdot (n_2 + 1) = 1.40 > q_1$, which means the backward condition is fulfilled. In the next step, the forward boundary condition will be tested to determine if the largest probability cell in the second PA needs to be moved to the first PA. We find $p_2^1 \cdot (n_2 - 1) = 0.3 < q_1$ which means the forward boundary condition is satisfied, and it is not required to move any cell in the second PA forward to the first PA. We keep testing as in Step 5 and obtain the optimal PAs as: $PA^*(1, 0.35, 1)$, $PA^*(2, 0.40, 3)$, $PA^*(3, 0.15, 3)$, and $PA^*(4, 0.1, 3)$ as the last row in Fig. 1. As a result, the *paging cost* under delay bound $\mathcal{D} = 4$, $E[C(4)]$ and the *average*

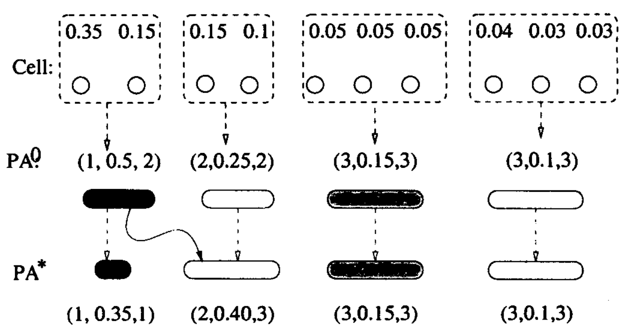

Fig. 1. An Example of Partition Algorithm and Paging Procedure.

delay, $E[D(4)]$ are computed as follows:

$$E[C(4)] = \sum_{i=1}^{4} q_i \cdot k_i = 1 \cdot 0.35 + 4 \cdot 0.40 + 7 \cdot 0.15 + 10 \cdot 0.10 = 4.0,$$

$$E[D(4)] = \sum_{i=1}^{4} i \cdot q_i = 1 \cdot 0.35 + 2 \cdot 0.40 + 3 \cdot 0.15 + 4 \cdot 0.10 = 2.0. \quad (12)$$

IV. Performance Analysis

The numerical results for uniform and non-uniform location probability distributions are provided in this section. We compare the *paging costs* and *average delays* of the proposed scheme with three other paging schemes: broadcast paging [7], [14], selective paging [1], and highest probability first (HPF) paging scheme [11].

The *paging costs* and *average delays* versus the delay bounds, $\mathcal{D}$, using (2) and (3) for the uniform distribution are shown in Fig. 2 and 3. For the selective paging scheme, we choose one case in which an LA is divided into three partitions with location probabilities 0.6, 0.2, and 0.2 [1]. Fig. 2 shows the *paging costs* $C(\mathcal{D})$ as a function of $\mathcal{D}$ for an LA with twenty cells (N=20) by using (2). It can be seen in Fig. 2 that the paging costs decrease with the increasing paging delay bounds for all paging schemes, except broadcast scheme [7], [14]. The *paging costs* of the optimal partition algorithm and HPF paging scheme fall very fast as the delay bound increases. Specifically, when the paging delay bound is 5, the paging costs achieve the small asymptotic value. The paging costs using selective paging scheme [1] remain the same after $\mathcal{D} = 3$ because there are three partitions. We observe that the optimal partition algorithm causes the minimum paging costs, which are the same as the theoretical result from HPF scheme [11]. It is also observed that the average delay increases as the delay bound increases. Nevertheless, we consider that the paging costs have higher priority than the average delays under the delay bounds. We conclude that the optimal partition algorithm produces the same performance as HPF when it can be applied to the uniform location distribution.

Next, we calculate the *paging costs* and *average delays* for non-uniform location probability distributions which may not be represented by a particular function. Two non-uniform distribution cases created randomly are considered in Table I. We demonstrate that the *paging costs* and *average delays* resulted from different paging schemes. The HPF scheme is not included in which the delay constraint is considered as a weighted factor in determining the minimum paging cost, is introduced in [11]. However, to find the optimal delay weighted factor, an analogous or corresponding continuous probability density function for a discrete probability distribution must be found. This is not a trivial problem for the non-uniform discrete distributions.

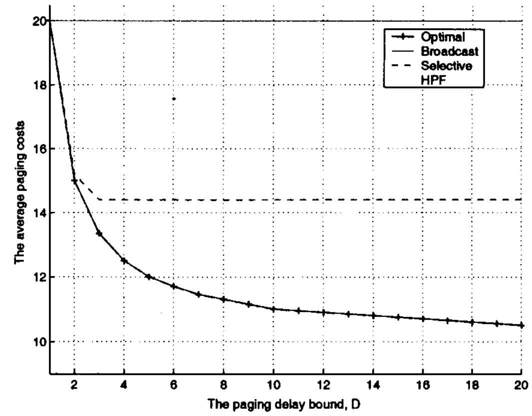

Fig. 2. The Paging Costs for Uniform Distribution.

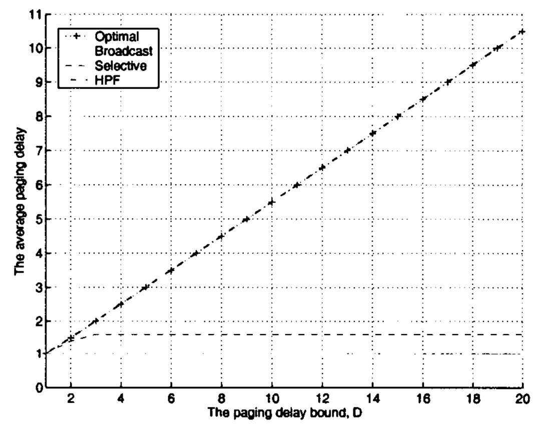

Fig. 3. The Average Delays for Uniform Distribution.

TABLE I
The Initial Location Probabilities.

Page Area	Sequence Number j	Location Probability p_j Case A	Case B	Number of Cells
PA_1	1	0.36	0.28	1
PA_2	2	0.31	0.26	1
PA_3	3	0.05	0.08	1
PA_4	4	0.05	0.08	1
PA_5	5	0.045	0.05	1
PA_6	6	0.045	0.05	1
PA_7	7	0.04	0.05	1
PA_8	8	0.04	0.05	1
PA_9	9	0.03	0.05	1
PA_{10}	10	0.03	0.05	1

The *paging costs* and *average delays* of case A and B are shown in Table II. The details of each PA, such as the paging sequence, the location probability, and the number of cells

TABLE II

THE COMPARISON OF PAGING COSTS AND DELAYS

Partitions (PAs)		$PA^*(i, q_i, n_i)$	$E[C(\mathcal{D})]$	$E[D(\mathcal{D})]$
Case A	Optimal	(1,0.36,1); (2,0.36,2): (3,0.28,7)	**4.24**	1.92
$\mathcal{D}=3$	Selective	(1,0.55,5): (2,0.31,1): (3,0.14,4)	6.01	*1.59*
Case A	Optimal	(1,0.36,1); (2,0.31,1): (3,0.19,4): (4,0.14,4)	**3.52**	2.11
$\mathcal{D}=4$	Selective	(1,0.55,5): (2,0.31,1): (3,0.14,4)	6.01	*1.59*
Case A	Optimal	(1,0.36,1); (2,0.31,1): (3,0.10,2): (4,0.13,3): (5,0.10,3)	**3.29**	2.03
$\mathcal{D}=5$	Selective	(1,0.55,5): (2,0.31,1): (3,0.14,4)	6.01	*1.59*
Case B	Optimal	(1,0.54,2); (2,0.26,4): (3,0.20,4)	**4.64**	1.66
$\mathcal{D}=3$	Selective	(1,0.62,3): (2,0.18,3): (3,0.20,4)	4.94	*1.58*
Case B	Optimal	(1,0.54,2); (2,0.16,2): (3,0.15,3): (4,0.15,3)	**4.27**	1.91
$\mathcal{D}=4$	Selective	(1,0.62,3): (2,0.18,3): (3,0.20,4)	4.94	*1.58*
Case B	Optimal	(1,0.54,2); (2,0.16,2): (3,0.10,2): (4,0.10,2): (5,0.10,2)	**4.12**	2.06
$\mathcal{D}=5$	Selective	(1,0.62,3): (2,0.18,3): (3,0.20,4)	4.94	*1.58*
Broadcast Scheme		(1,1,10)	10	1

under $\mathcal{D}=3,\ 4,\ 5$, are also illustrated. The paging costs are indicated in bold and the average delays are indicated in italics in Table II. The paging costs decrease as the delay bound increases. For instance, the paging costs are changed from $4.24 \rightarrow 3.52 \rightarrow 3.29$ when $\mathcal{D}$ is changed from 3 to 4 and to 5. It can be seen that the optimal paging scheme results in the minimum paging costs when $\mathcal{D}$ is equal to 3, 4, and 5, respectively. The selective paging scheme produces small average paging delays. According to the *paging costs* of case A, the paging costs of the optimal partition algorithm can be reduced up to 40% compared with that of the selective paging scheme. We also notice that the paging costs using optimal scheme are very small even though $\mathcal{D}$ is small such as 3. In addition, the paging costs do not change very fast when delay bounds reach a certain value. For example, in case A, the paging cost is reduced up to $\mathcal{D}=4$. After that, the improvement of paging cost is not as sensitve as for small values of $\mathcal{D}$. Because of this, the delay bounds can be applied to locate the MT while minimizing the paging costs. The selective paging scheme also causes good results in some cases, but it needs to group the cells into PAs using complex computation; furthermore, it depends on the size and shape of the LA, which makes it not flexible. Our proposed scheme is simple to implement and definitely reduces the paging costs significantly for various probability distributions.

V. CONCLUSION

We have introduced an optimal paging scheme which is applicable to different location probability distributions in wireless systems. This scheme is simple to implement and it is able to minimize the paging costs under delay bounds, in particular, when the location probability distribution is nonuniform, which is not considered in many paging schemes. We also carried out some experiments and our results revealed that our proposed scheme is still very effective in minimizing the paging costs. On the whole, the new proposed scheme in this paper provides an optimal, generic, and feasible method of paging process in wireless systems.

REFERENCES

[1] A. Abutaleb and V.O.K. Li, "Paging Strategy Optimization in Personal Communication System," *ACM-Baltzer Journal of Wireless Networks (WINET)*, Vol. 3, pp. 195-204, August 1997.

[2] I.F. Akyildiz, H.S.M. Ho, and Y-B. Lin, "Movement-Based Location Update and Selective Paging for wireless Networks," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, pp. 629-638, August 1996.

[3] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioğlu and W. Wang, "Mobility Management in Next-Generation Wireless Systemes," *Proceedings of the IEEE*, Vol. 87, pp 1347-1384, August 1999.

[4] B-N. Amotz, I. Kessler and M. Sidi, "Mobile Users: To Update or not to Update?," *IEEE INFOCOM'94*, Vol. 2, pp. 570-576, June 1994.

[5] L.P. Araujo and J.R.B.de Marca, "A Comparative Analysis of Paging and Location Update Strategies for PCS Networks," *IEEE ICC'98*, Vol. 3, pp. 1395-1399, June 1998.

[6] A. Bhattacharya and S. K. Das, "LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks," *Proc. of ACM/IEEE MobiCom'99*, pp. 1-12, August 1999.

[7] H.C. Lee and J. Sun, "Mobile Location Tracking by Optimal Paging Zone Partitioning," *IEEE ICUPC'97*, Vol. 1, pp. 168-172, October 1997.

[8] T. Liu, P. Bahl and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 16, No. 6, pp. 922-936, August 1998.

[9] Z. Naor and H. Levy, "Minimizing the Wireless Cost of Tracking Mobile Users: An Adaptive Threshold Scheme," *IEEE INFOCOM'98*, pp 720-727, March 1998.

[10] C. Rose, "State-Based Paging/Registration: A Greedy Technique," *IEEE Transaction Vehicular Technology*, Vol. 48, No. 1, pp 166-173, January 1999.

[11] C. Rose and R. Yates, "Minimizing the Average Cost of Paging Under Delay Constraints," *ACM-Baltzer Journal of Wireless Networks (WINET)*, Vol. 1, pp. 211-219, February 1995.

[12] G. Wan and E. Lin, "Cost Reduction in Location Management Using Semi-Realtime Movement Information," *ACM-Baltzer Journal of Wireless Networks (WINET)*, Vol 5, No. 4, pp 245-256, 1999.

[13] T.Y.C. Woo, T.F.L. Porta, J. Golestani and N. Agarwal, "Update and Search Algorithms for Wireless Two-Way Messaging: Design and Performance," *IEEE INFOCOM'98*, pp. 737-747, March 1998.

[14] H. Xie, S. Tabbane and D.J. Goodman, "Dynamic Location Area Management and Performance Analysis," *42nd IEEE VTC'92*, pp. 536-539, May 1992.