Department of Electrical and Computer Systems Engineering

Technical Report MECSE-19-2003

Shift Comparison Algorithm for Minimising Paging Costs under Delay Bounds

Hongmin Chen, N. Mani and B. Srinivasan

MONASH UNIVERSITY

Shift Comparison Algorithm for Minimising Paging Costs under Delay Bounds

H. Chen, N. Mani and B. Srinivasan

A shift comparison algorithm is proposed to minimise the paging cost of location management in cellular wireless networks. Numerical results demonstrate that this paging method performs better than other methods for both uniform and non-uniform location probability distributions.

Introduction: Location management is a key issue in cellular mobile networks and personal communications services (PCS). It is concerned with those network functions that are necessary to track mobile users or terminals wherever they are in the network coverage area. The network service area is divided into many location areas (LAs) with each LA consisting of a number of cells. The two fundamental operations for locating a mobile terminal in cellular network are location update and paging. The number of the cells being paged to locate a called mobile terminal (MT) determines the traffic that passes through the network. The paging cost is related to the efficiency of bandwidth utilization, and it is measured in terms of cell to be polled before the called mobile terminal is found [1].

Based on certain mobile models and calling patterns, location probabilities of cells have been used to cut the paging cost [2, 3]. In selective paging schemes, the LA was divided and lined into a sequence of partitions areas (PA) with each PA consisting of a cluster of cells. The PAs were searched by their positions in the sequence. In [4], the location probabilities of cells of the PAs were proved to be in decreasing order. In [4, 5], backward and forward boundary conditions were used to determine the number of cells in the neighbouring PAs along the sequence. However the paging cost can still be further minimised. In this work, a shift comparison algorithm (SCA) is proposed. All PAs of the paging sequence were incorporated into the calculation of the minimum paging cost.

Numerical results demonstrate that SCA performs better over various location probability distributions.

Algorithm formulation: We assume that a LA consists of N cells, $c_1, c_2, \ldots, c_N$. The probabilities of finding the called MT in the cells are, $p_1, p_2, \ldots, p_N$. With a condition of paging delay bound D, the cells shall be grouped into D paging areas PA_1, PA_2 ,…, PA_D and searched sequentially. The average paging cost of this paging sequence can be expressed as [1]

$$E[C_{EA}(D)] = \sum_{i=1}^{D} s_i q_i \qquad (1)$$

where $q_i = \sum_{c_k \in PA_i} p_k$ and $s_i = \sum_{k=1}^{i} n_k$ where n_i is the number of cells of PA_i.

From the property of the paging sequence, which has the minimum average paging cost, four important lemmas are derived, and in SCA, they are used to identify the cells in the PAs.

Lemma 1: For a paging sequence PA_1, PA_2, …, PA_D, which has the minimum average paging cost, if cells, $c_{front} \in PA_{front}$, $c_i \in PA_i$; where PA_{front} is in front of PA_i in the paging sequence, then $p_{front} \geq p_i$. In other words, the sequence PA_1, PA_2, …, PA_D has cells in descending order of probabilities[4, 5].

Lemma 2: The numbers of cells of the PAs, $n_1, n_2, \ldots, n_D$, satisfy $n_1 \leq n_2 \leq \ldots\ldots \leq n_D$.

Proof of lemma 2: Suppose there exists PA_i and PA_{i+1} for which $n_i > n_{i+1}$. If we move the last cell of PA_i, $c_{i,last}$ to PA_{i+1}, then we get a new paging sequence. In this new sequence, the number of cells of PA_i will be $n_i - 1$. The change of the average paging cost will be $(n_{i+1})p_{i,last} - \sum_{k=1}^{n_i - 1} p_{i,k}$. Since $n_i - 1 \geq n_{i+1}$, from lemma 1, this change is negative. The average paging cost of this new sequence is less than the original one. This is contrary to the assumption that the original paging sequence has the minimum average paging cost. Therefore $n_i \le n_{i+1}$.

Lemma 3: Any subsequence PA_i, PA_{i+1} ,…, PA_j, which is a section of PA_1, PA_2,…, PA_D, has the minimum average paging cost for the cells in it.

Proof of lemma 3: Suppose a paging sequence PA'_i, PA'_{i+1} ,…, PA'_j has a lower paging cost than PA_i, PA_{i+1} ,…, PA_j and these two sequences have the same set of cells. If we replace PA_i, PA_{i+1} ,…, PA_j with PA'_i, PA'_{i+1} ,…, PA'_j in PA_1, PA_2 ,…, PA_D, then we get a new sequence, PA_1 ,…, PA_{i-1}, PA'_i ,…, PA'_j , PA_{j+1},…,PA_D and the average paging cost of this new sequence will be less than the original one. Again this is contrary to the assumption that PA_1, PA_2 ,…, PA_D has the minimum average paging cost. Therefore lemma 3 holds.

Lemma 4: If PA_1, PA_2,…, PA_D has the minimum average paging cost for N-1 cells and $PA`_1$, $PA`_2$,…, $PA`_D$ has the minimum average paging cost for N cells, the same N-1 cells and the $N^{th}$ cell which has the lowest location probability, then $PA`_D$ can only have one more cell than PA_D.

Proof of lemma 4: If the N^{th} cell is taken out of PA'_D, the average paging cost of the $PA`_1$, $PA`_2$,…, $PA`_D$ will be reduced by $\sum_{c_i \in (PA'_D - c_N)} p_i + N \times p_N \quad c_i \ne c_N$, on the other hand if the N^{th} cell is placed into PA_D, the average paging cost of the PA_1, PA_2, …, PA_D will be increased by $\sum_{c_j \in PA_D} p_j + N \times p_N$.

From lemma 1, if PA'_D has more than one cell than PA_D, the increase will be less than the reduction. Furthermore PA_1, PA_2 ,…, PA_D has the minimum average paging cost for N-1 cells. Thus if we place the N^{th} cell into PA_D, then we get a new paging sequence and the average paging cost of this new sequence will be less than the cost of $PA`_1$, $PA`_2$,…, $PA`_D$. This is contrary to the assumption that $PA`_1$, $PA`_2$ ,…, $PA`_D$ has the minimum paging cost. Therefore $PA`_D$ can only have one more cell than PA_D.

The Shift Comparison Algorithm: Initially cells are sorted in descending order of their location probabilities, $p_1 > p_2 > ... > p_{N-1} > p_N$. Then the first D *cells*, $c_1, c_2, ..., c_D$, are placed into the buffers of $PA_1, PA_2, ..., PA_D$ individually. Thus the probabilities of PAs, $q_1, q_2, ..., q_D$ equal $p_1, p_2, ..., p_D$ respectively.

Step 1: The first one of the remaining N-D *cells* is selected and placed into the last position of the buffer of PA_D. So we get a paging sequence for $D+1$ cells. The average paging cost of the paging sequence is calculated and put into a temporary variable as the minimum average paging cost.

Step 2: A shift procedure is processed along the cells buffers of PAs. Each time the first cell of a PA is shifted to its front neighbour in the paging sequence.

Step 3: If after a shift from PA_i to PA_{i-1}, PA_{i-1} has less number of cells than PA_i, then a new average paging cost is calculated and is compared with the temporary variable. If the calculated average paging cost is smaller, the temporary variable will be updated with the new value and the shift position is recorded. Step 2 and Step3 are repeated until the shifting reaches PA_1.

Step 4: Suppose the recorded position is PA_j. We recovered $PA_1, ..., PA_j$ and update the probabilities of $PA_j, PA_{j+1}, ..., PA_D$,

$q_j = q_j + p_{j+1, 1}$ where $p_{j+1, 1}$ is the probability of the first cell in the buffer of PA_{j+1};

$q_{j+1} = q_{j+1} - p_{j+1, 1} + p_{j+2, 1}$; …… ; $q_{D-1} = q_{D-1} - p_{D-1, 1} + p_{D, 1}$;

$q_D = q_D - p_{D, 1} + p_{new}$ where p_{new} is the probability of the new coming cell;

and the number of cells of PA_j, $n_j = n_j + 1$;

The steps 1 to 4 are repeated until all the N-D cells are processed. In step 3, if PA_{i-1} has the same number of cells of PA_i, the average paging cost of the sequence, which is created by the shifting, is not computed and compared. So if and only if PA_{i-1} has one cell less than PA_i along the paging sequence, the times of calculations and comparisons will reach the maximum and it satisfies $\frac{(times+1)\times times}{2} = i$; where i means the i^{th} cycle of the processes. *Because* the times must be a positive number, $times = \left\lceil (-1+\sqrt{8\times i+1}) / 2 \right\rceil$. Thus the computation complexity of SCA is Θ(N). It shows that SCA is a feasible paging algorithm.

Test results and comparison: Firstly, we compared SCA with the "optimal" paging scheme described in [4]. Two types of data were derived from [4] and [5]. In case A, the location probabilities of cells are: 0.35, 0.15, 0.15, 0.1, 0.05, 005, 0.05, 0.04, 0.03 and 0.03. The total number of cells N is 10, and the paging delay bound D is 4. The paging sequences and the average paging costs for these two algorithms are shown in Table 1. In case B, the location probabilities of cells are: 0.28, 0.26, 0.08, 0.08, 0.05, 0.05, 0.05, 0.05, 0.05 and 0.05. The paging delay bound D is 5. Results are shown in Table 2. Table 1 and 2 show than the average paging costs given by the optimal paging scheme in [4] and [5] can be further minimised by SCA.

To show the performance of SCA under different types of location probability distributions, we compared SCA with the three other schemes described in [6]. They are Reverse, Semi-reverse and Uniform paging schemes, and they are designed to meet different performance requirements [6]. Typically, results for a truncated Gaussian distribution and an exponential distribution are shown in figure 1 and 2. SCA performs better than all the three other schemes. This complies with the design rules of the lemmas 1 to 4.

Conclusion: In this letter, we have presented an effective paging scheme that is capable of minimising the average paging cost under delay bounds. It is a simple scheme, which is easy to implement in wireless systems. The performance of the scheme is analyzed with numerical data. The results show an improved average paging cost compared to existing schemes.

[1] C. Rose and R. Yates, "Minimizing the average cost of paging underdelay constraints," *Wireless Networks*, vol. 1, pp. 211-219, February 1995.
[2] A. Abutaleb and V. O. K. Li, "Paging strategy optimization in personal communication system," *Wireless Networks*, vol. 3, pp. 195-204, August 1997.
[3] 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*, vol. 16, pp. 389-400, August 1998.
[4] Wenye Wang, I. F. Akyildiz, and G. L. Stuber, "Optimal Partition Algorithm for Minimization of Paging Costs," presented at GLOBECOM '00. IEEE, 27 Nov.-1 Dec. 2000.
[5] Wenye Wang, I. F. Akyildiz, and G. L. Stuber, "An Optimal Paging Scheme for Minimization Signaling Costs Under Delay Bounds," *IEEE communications letters*, vol. 5, pp. 43-45, February 2001.
[6] Wenye Wang, I. F. Akyildiz, and G. L. Stuber, "Effective Paging Schemes with Delay Bounds as QoS Constraints in Wireless Systems," *Wireless Networks*, vol. 7, pp. 455-466, 2001.

Authors' affiliations:

H. Chen and N. Mani (Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Vic 3800, Australia); and B. Srinivasan (School of Computer Science and Software Engineering, Monash University, Clayton, Vic 3800, Australia)

E-mail: hongmin.chen@eng.monash.edu.au

Table 1. The comparison of average paging cost of case A
Table 2. The comparison of average paging cost of case B

Figure 1. Comparison under Truncated Gaussian Distribution
Figure 2. Comparison under Exponential Distribution

Table 1

Case A		PA_1	PA_2	PA_3	PA_4	*Average Paging Cost*
SCA	Probability	0.35	0.3	0.2	0.15	3.95
	Number of Cells	1	2	3	4	
"Optimal"	Probability	0.35	0.4	0.15	0.1	4.0
	Number of Cells	1	3	3	3	

Table 2

Case A		PA_1	PA_2	PA_3	PA_4	PA_5	*Average Paging Cost*
SCA	Probability	0.28	0.26	0.16	0.15	0.15	3.99
	Number of Cells	1	1	2	3	3	
"Optimal"	Probability	0.54	0.16	0.1	0.1	0.1	4.12
	Number of Cells	2	2	2	2	2	

Figure 1

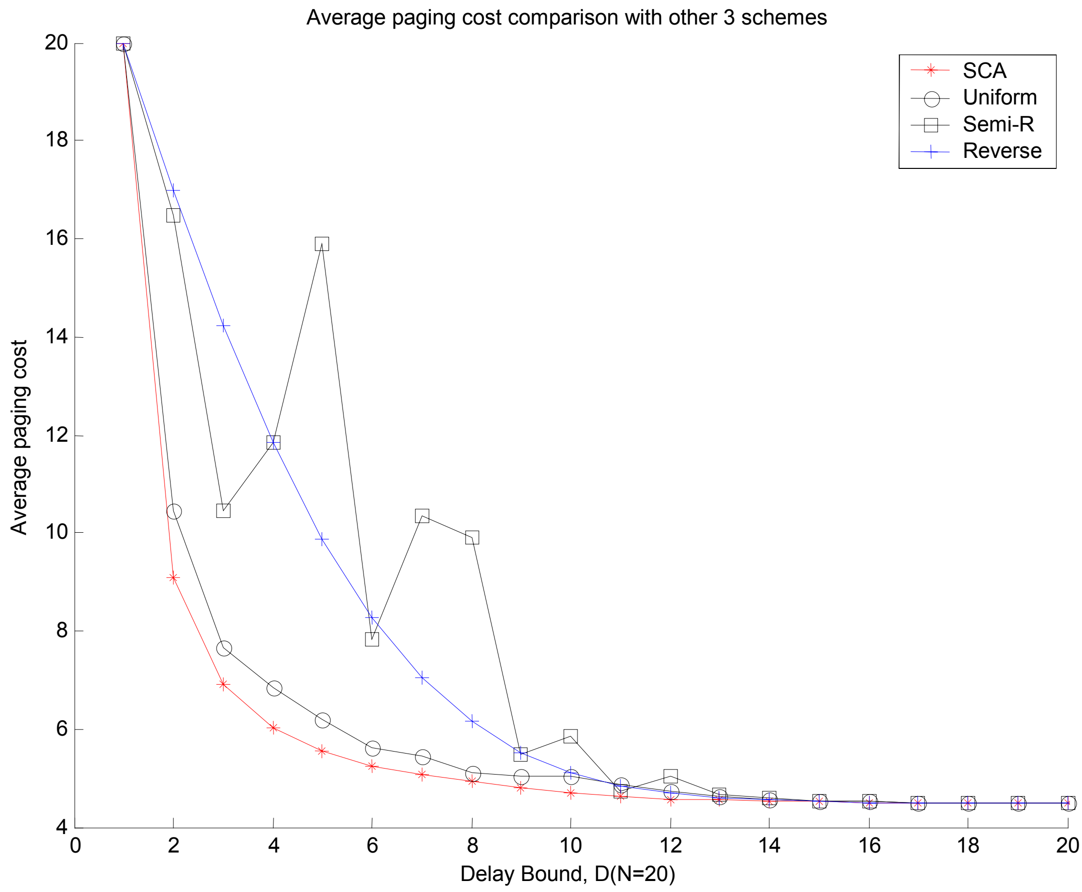
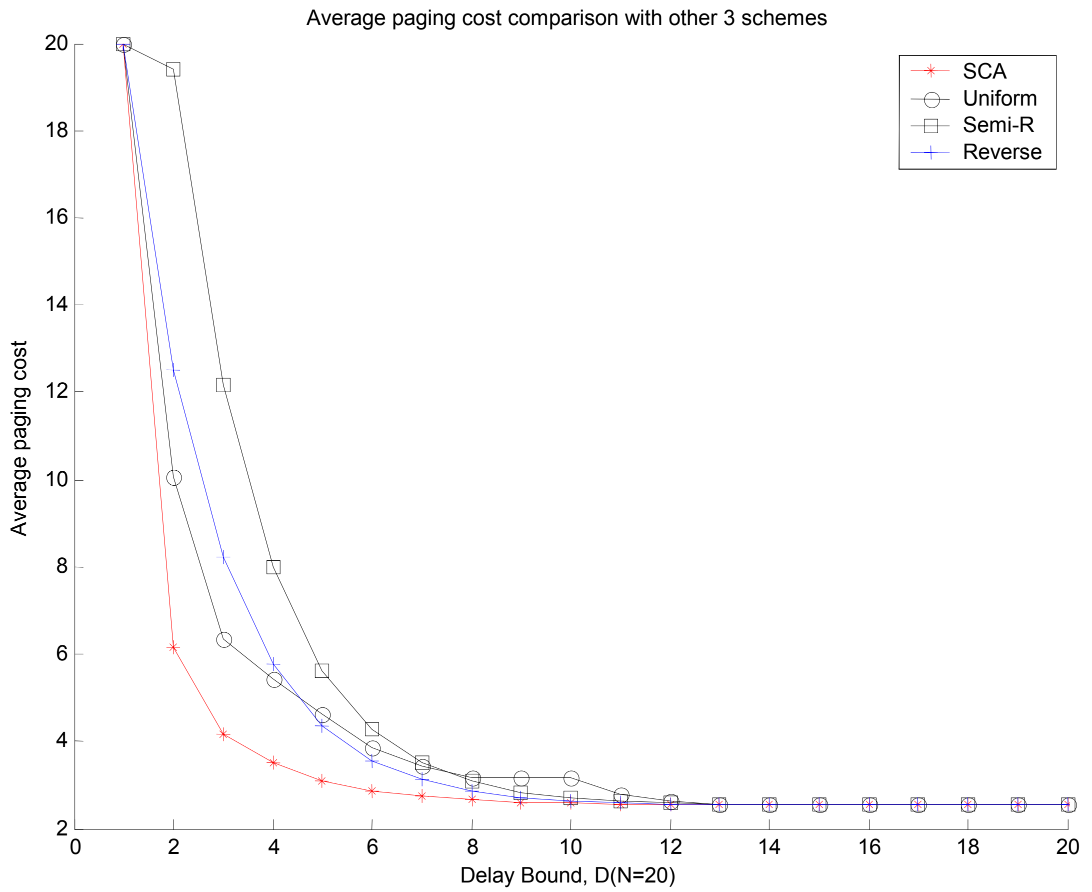

Average paging cost comparison with other 3 schemes
SCA
Uniform
Semi-R
Reverse
Average paging cost
Delay Bound, D(N=20)
20
18
16
14
12
10
8
6
4
0
2
4
6
8
10
12
14
16
18
20

Figure 2


Average paging cost comparison with other 3 schemes
SCA
Uniform
Semi-R
Reverse
Average paging cost
Delay Bound, D(N=20)
0
2
4
6
8
10
12
14
16
18
20