Intelligent Paging Strategy based on Location Probability of Mobile Station and Paging Load Distribution in Mobile Communication Networks

Dong-Jun Lee  
Mobile Development Team,  
Telecommunication Systems Division,  
Information & Communication Business,  
Samsung Electronics Co., Ltd.,  
Republic of Korea

HyeJeong Lee and Dong-Ho Cho  
Div. of Electrical Engineering,  
Dept. of Electrical Engineering and Computer Science,  
Korea Advanced Institute of Science and Technology,  
Republic of Korea

Abstract—When the cells in a location area are paged sequentially based on the specific information such as last registered area or mobile speed, paging load may be non-uniformly distributed among the cells. This non-uniform paging traffic causes additional paging delay due to the increase of waiting time in the cell with high paging load. In this paper, we introduce a new paging strategy in which the paging sequence in a location area is optimized based on both the location probability of a mobile terminal and paging load distribution among the cells. From numerical results, we show that the paging cost of our proposed scheme is very close to the optimal one. In addition, we observe the effects of the paging load distribution and the location probability distribution on the paging cost.

I. INTRODUCTION

Third generation mobile communication systems such as UMTS and cdma2000 are characterized as offering worldwide communication and various multimedia services. In the mobile networks, it is very critical to keep track of the location of a mobile terminal when it moves between cells or even between different systems. In addition, increasing demands on universal access and global roaming make efficient location (mobility) management play a very important role.

In general, location management consists of two basic procedures: location registration (location update) and paging. Location registration is the process in which a mobile user informs the networks of the current location information. To facilitate this, the entire service area of the communication system is divided into several location areas and a location update is performed when a mobile user crosses a boundary of the location area. When a call request arrives to the user, paging is performed within the last registered location area to search for the mobile user by sending polling messages to the cells.

In particular, several paging schemes such as blanket paging, sequential paging, and intelligent paging, have been proposed for call delivery. Blanket paging scheme, in which all cells in a location area are paged simultaneously to deliver an incoming call, causes low paging delay. However, blanket paging wastes excessive wireless resources. In sequential paging, a location area is divided into smaller areas called a paging area and the group of cells in a paging area is searched in one polling cycle. A polling cycle in a sequential paging scheme is defined by the round trip time from the time when a paging message is transmitted to the time when the response is received. Sequential paging scheme is efficient to reduce paging load, but it may increase paging delay exponentially.

During the last decade, much research has been done on an intelligent paging scheme, which determines an efficient paging sequence in a multi-step paging [1]–[4]. To improve the performance of paging strategy in intelligent paging, cells are paged sequentially based on specific user information such as last interaction area, mobile speed and etc. Because cells with higher probability in which the target user may be found should be paged first to reduce paging traffic and paging delay, previous research mainly has focused on the calculation of the user location probability. User location probability can be determined based on most recent location of registration, user mobility and elapsed time since the last location update.

A multi-step paging based on user location probability is useful for finding the target user. However, in contrast to a blanket paging in which all cells in a location area have the same paging load, a multi-step paging searches cells selectively and so paging traffic is not distributed uniformly among the cells of the location area. Non-uniform paging load distribution among cells makes paging response times in cells different each other. Moreover, if paging in a particular paging area fails, the paging delay of the paging area is determined by the longest response time, which is derived from that cell in the paging area that has the greatest paging load. Therefore, to optimize the paging sequence with respect to the paging load and the paging delay, the non-uniform paging load distribution should be considered in addition to the location probability distribution. While the location probability distribution is related to the individual mobility behavior of a mobile terminal, the paging load distribution is determined by the global distribution of traffic in the system.

In this paper, we propose an enhanced paging scheme
considering both the location probability distribution of a
mobile terminal and the non-uniform paging load distribution.
The rest of this paper is organized as follows. In Section II,
we provide an analytic model and formulate the paging cost.
In Section III, proposed strategy based on both the paging load
and the user location probability is explained in detail. We
show numerical results in Section IV and conclude this paper
in Section V.

II. PAGING COST FORMULATION

In this section, we define modified paging cost formula
similar to the approach adopted in [5]. In our model, we
additionally consider the effect of paging load distribution
on the paging cost.

We assume that there are \( N_c \) cells in a location area. Let
\( \lambda_i \) be the probability that a user exists in the \( i \)th cell and \( \rho_i \)
be the paging load, which is the number of paging requests
waiting in the \( i \)th cell (\( i = 1, 2, \ldots N_c \)), respectively. Then,
\( \sum_{i=1}^{N_c} \lambda_i = 1 \) (\( 0 \leq \lambda_i \leq 1 \)) and \( \rho_i \) may be given as an instant
value when a paging process is initiated or as a short-term or
long-term average of paging load in a system. As shown in
Fig. 1, it is assumed that a location area is divided as \( N_p \)
 paging areas (\( 1 \leq N_p \leq N_c \)) for a multi-step paging, and the
size of the \( j \)th paging area (i.e., the number of cells in the \( j \)th
 paging area) is \( n_j \) (\( j = 1, 2, \ldots N_p \)).

We define the paging success probability of a paging area as
the probability that the target user for an incoming call will be
found in the corresponding paging area. Then, paging success
probability of the \( j \)th paging area, \( p_j \), can be calculated as

\[
p_j = \sum_{i=m_j+1}^{n_j} \lambda_i,
\]

(1)

where \( m_j = \sum_{i=1}^{j-1} n_i \), \( m_1 = 0 \) and \( \sum_{j=1}^{N_p} p_j = 1 \).

Now, we derive the paging delay cost. In order to calculate
the paging delay that is the time taken until paging is
successful, we assume that paging request is queued and not
blocked although paging load in a cell is high. Let \( \hat{D}_j \) be the
normalized paging delay when the paging in the \( j \)th paging area
succeeds and \( \bar{D}_j \) be the normalized delay when the paging
in the \( j \)th paging area fails. Then, average paging delay (\( D \)) of
a mobile user until paging succeeds is calculated in sequential
way as follows.

\[
D = p_1 \hat{D}_1 + (1-p_1) \left[ \hat{D}_1 + \frac{p_2 \hat{D}_2}{1-p_1} \right] + \cdots \left( 1 - \sum_{j=1}^{N_p-1} p_j \right) \left[ \hat{D}_{N_p-1} + \frac{p_{N_p} \hat{D}_{N_p}}{1 - \sum_{j=1}^{N_p-1} p_j} \right] = \sum_{j=1}^{N_p} p_j \hat{D}_j + \sum_{j=1}^{N_p} \left[ 1 - \sum_{i=1}^{j} p_i \right] \cdot \hat{D}_j
\]

(2)

In (2), we can obtain the average delays \( \hat{D} \) and \( \bar{D} \) which are
caused by successful paging and failed paging respectively, as
follows.

\[
\hat{D} = \sum_{j=1}^{N_p} p_j \hat{D}_j
\]

(3)

\[
\bar{D} = \sum_{j=1}^{N_p-1} \left[ 1 - \sum_{i=1}^{j} p_i \right] \cdot \bar{D}_j
\]

(4)

Given that the transmission time for a paging message over
a wireless link, \( 1/\tau \), is assumed to be constant throughout
cells, the normalized paging delay of successful paging in \( j \)th
 paging area, \( \hat{D}_j \), is represented by

\[
\hat{D}_j = \frac{1}{p_j} \sum_{i=m_j+1}^{n_j} \lambda_i \cdot \left( \frac{\rho_i}{\tau} \right),
\]

(5)

where \( \rho_i/\tau \) is the normalized paging load in each cell. Meanwhile,
the normalized paging delay of failed paging in
\( j \)th paging area, \( \bar{D}_j \), is expressed as

\[
\bar{D}_j = \frac{\max(\rho_{m_j+1}, \rho_{m_j+2}, \ldots, \rho_{m_j+n_j})}{\tau}
\]

(6)

Note that from (3) and (5), \( \bar{D} \) is determined by the given
location probability distribution \( \lambda_i \) and paging load distribution
\( \rho_i \), regardless of the particular paging sequence.
Therefore, hereafter, we consider only \( \hat{D} \) as the paging delay
cost. Because \( \sum_{j=1}^{N_p} p_j = 1 \), \( \hat{D} \) in (4) can be rewritten as

\[
\hat{D} = \sum_{j=1}^{N_p-1} \left[ 1 - \sum_{i=1}^{j} p_i \right] \hat{D}_j = \sum_{j=1}^{N_p} \left[ 1 - \sum_{i=1}^{j} p_i \right] \cdot \hat{D}_j
\]

(7)
Similarly to (2), the paging load cost, which is the average number of cells searched until paging succeeds, is defined by

\[ L = \sum_{j=1}^{N_p} \left( p_j \cdot \sum_{i=1}^{j} n_i \right). \]  

(8)

Accordingly, the total paging cost, \( C \), can be given by the weighted sum of the paging delay cost (\( \hat{D} \)) and the paging load cost (\( L \)) as follows.

\[
C = L + \omega \cdot \hat{D}
\]

\[
= \sum_{j=1}^{N_p} \left( p_j \sum_{i=1}^{j} n_i \right) + \omega \sum_{j=1}^{N_p} \left[ \left( 1 - \sum_{i=1}^{j} p_i \right) \cdot \hat{D}_j \right]
\]

\[
= \sum_{j=1}^{N_p} \left[ p_j \sum_{i=1}^{j} n_i + \omega \left( 1 - \sum_{i=1}^{j} p_i \right) \cdot \hat{D}_j \right]
\]

(9)

where \( \omega \) is a weighting factor which is called delay factor. According to the constraints of the amount of paging traffic and the delay bound, the delay factor \( \omega \) can be appropriately assigned.

### III. PROPOSED INTELLIGENT PAGING SCHEME

In [4], a paging scheme that optimally partitions a location area into several paging areas has been proposed, but the optimal partitioning scheme is based only on the user location probability, without consideration of the paging load distribution. In this paper, we propose an enhanced paging scheme that divides a location area into paging area by considering both paging load distribution and location probability distribution.

First, we sort all \( N_c \) cells in the increasing order of \( \rho_j / \lambda_i \) \((i = 1, 2, \cdots, N_c)\) and choose the sorted sequence for the initial paging sequence of our paging strategy. To increase paging success probability, a cell with a high location probability should be searched first. In addition, searching a cell with low paging load results in the reduction of paging response time. If paging process fails in a particular cells, it takes time to report its paging failure, thus, the overall paging delay will increase. Accordingly, proposed initial paging sequence according to increasing order of \( \rho_j / \lambda_i \) is feasible and very reasonable.

Following the determination of the initial paging sequence, we find optimal sizes of paging areas and allocate cells to given sets of paging areas. Given that there are \( N_c \) cells in a location area, the location area can be divided into paging areas with various sizes. Fig. 2 shows an example to partition \( N_c \) cells into paging areas by the typical branch-and-bound method and our proposed scheme. In Fig. 2, the number within a circle represents a size of the paging area \((n_j)\) (i.e., the number of cells allocated in the paging area) and the leftmost paging area will be searched first in the paging process.

Let state \( n \) be all possible cases that partition \( n \) cells. If the size of the paging area that will be paged first is \( k \), there are \( 2^{n-k-1} \) possible partitions of the remaining \((n-k)\) cells. These \( 2^{n-k-1} \) possible sets of paging area sizes correspond to state \( n-k \). Note that in Fig. 2 (b), state 2 includes state 0 and 1, and state 3 includes state 0, 1 and 2, respectively.

Accordingly, a location area with \( n \) cells is divided into a first paging area with size \( k \) \((k = 1, 2, \cdots, n)\) and state \( n - k \). In addition, state \( n \) can be divided into combinations of the first paging area with the size \( k \) and the remaining state \( n-k \), and can be generally represented by state 1 through state \( n-1 \), as shown in Fig. 2 (b).

If the size of each paging area is determined, all cells in a location area can be partitioned into each paging area in the sorted order for the initial paging sequence. Moreover, in Fig. 2 (b), if we calculate the optimal cell partitioning of state \( i \) \((i = 1, \cdots, n-1)\), the optimal cell partitioning for state \( n \) can be obtained recursively. If the size of the first paging area in state \( n \) is \( k \), the paging cost for state \( n \), \( C_n \), can be expressed as

\[
C_n = \left[ \left( \sum_{i=1}^{k} \lambda_i \right) \cdot k + \omega \cdot \left( 1 - \sum_{i=1}^{k} \lambda_i \right) \cdot \hat{D}(k) \right] + C_{n-k},
\]

for \( k = 1, \cdots, n-1 \)

(10)

where \( \hat{D}(k) \) is the maximum normalized load among \( k \) cells in the first paging area. In (10), the value of \( k \) that gives the minimum \( C_n \) is the optimal size of the first paging area for state \( n \). If the size of the first paging area is determined, the remaining cells in state \( n-k \) can be partitioned in a similar way to minimize the paging cost.

In the case of the conventional branch-and-bound method, to find an optimal partition for \( n \) cells into paging areas, \( 2^{n-1} \((= 1 + 1 + 2 + 4 + \cdots + 2^{n-1})\) paging sequences should be compared as shown in Fig. 2 (a). In contrast, we can reduce the total number of sets for paging areas to \((n^2 - n + 2)/2 \((= 1 + 1 + 2 + 3 + \cdots + (n-1))\) using the proposed partitioning algorithm, as shown in Fig. 2 (b). Moreover, our proposed algorithm requires time complexity \( O(n^2) \), which is far simpler than the exponential time complexity of the branch-and-bound method.
### Table I

**RATIO OF PAGING COST FOR THE PROPOSED METHOD TO THE PAGING COST OF THE OPTIMAL SEQUENCE**

<table>
<thead>
<tr>
<th>$\rho_{\text{dev}}$</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_C$</td>
<td>1.0000</td>
<td>1.0025</td>
<td>1.0043</td>
<td>1.0109</td>
<td>1.0054</td>
<td>1.0135</td>
</tr>
</tbody>
</table>

(a) when $\rho_{\text{dev}}$ varies ($N_c = 10$, $\omega = 1$, $\rho_{\text{mean}} = 10$, $\lambda_{\text{dev}} = 0.5$)

<table>
<thead>
<tr>
<th>$\rho_{\text{dev}}$</th>
<th>0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.4</th>
<th>0.5</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_C$</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

(b) when $\lambda_{\text{dev}}$ varies ($N_c = 10$, $\omega = 1$, $\rho_{\text{mean}} = 10.0$, $\rho_{\text{dev}} = 5.0$)

<table>
<thead>
<tr>
<th>$\rho_{\text{dev}}$</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_C$</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

(c) when $N_c$ varies ($\omega = 1$, $\rho_{\text{mean}} = 10.0$, $\rho_{\text{dev}} = 5.0$, $\lambda_{\text{dev}} = 0.5$)

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_C$</td>
<td>1.0064</td>
<td>1.0009</td>
<td>1.0005</td>
<td>1.0004</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

(d) when $\omega$ varies ($N_c = 10$, $\rho_{\text{mean}} = 10.0$, $\rho_{\text{dev}} = 5.0$, $\lambda_{\text{dev}} = 0.5$)

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, we provide numerical results for our paging strategy, on the basis of the analysis in Section II.

Suppose that the location probability ($\lambda_i$) of a user in $i$th cell is obtained from a Gaussian distribution with a mean ($\lambda_{\text{mean}}$) of 1.0 and a standard deviation of $\lambda_{\text{dev}}$. We normalize $\lambda_i$ with $\lambda_{\text{dev}}$ so that the sum of location probabilities of all cells in a location area is equal to 1. If the standard deviation ($\lambda_{\text{dev}}$) is 0, all cells have the same location probability. We assume that each cell has a normalized paging load ($\rho_i/\tau$), and $\rho_i/\tau$ is given by a Gaussian distribution with mean ($\rho_{\text{mean}}$) 10.0 and standard deviation $\rho_{\text{dev}}$. When $\rho_{\text{dev}}$ is 0, all cells have the same paging load. If a random number below zero is generated from the Gaussian distribution, we assign an arbitrary small positive value instead.

In Table I, numerical results of our proposed partitioning algorithm and the conventional branch-and-bound method are compared. Here, $R_C$ denotes the ratio of the paging cost for our proposed scheme to the optimal paging cost obtained by using the branch-and-bound method. We can see that the paging costs of our proposed strategy using the heuristic formula that considers the effect of non-uniform paging load distribution on paging delay as well as the effect of typical location probability distribution. In addition, we have introduced a simple polynomial time algorithm to determine the sub-optimal paging sequence. Numerical results show that the paging sequence obtained by our proposed scheme is almost equivalent to the optimal paging sequence. Moreover, because our proposed scheme requires very low computation complexity, it can be efficiently used in real-time paging applications.

<table>
<thead>
<tr>
<th>REFERENCES</th>
</tr>
</thead>
</table>

IEEE Communications Society  
0-7803-8533-0/04/$20.00 (c) 2004 IEEE
Fig. 3. Effects of the deviation of the paging load ($\rho_{dev}$)

Fig. 4. Effects of the deviation of the location probability ($\lambda_{dev}$)

