

Reducing Location Registration Cost in Mobile Cellular Networks

Ki Ho Seo and Jang Hyun Baek

Mobility management is important in mobile cellular networks. In this study, we considered an enhanced location-based registration (ELR) method. In the ELR method, even when a mobile phone enters a cell to find that the cell is already on its list (of visited cells) and then updates its main counter, it does not remove any cells from the list (memory space permitting), which gives better performance than the location-based registration (LR) method. However, the location registration cost of the ELR method is still high, and there is a lot of room for improvement with regards to this matter. We now propose an improved version of the ELR method; namely, the improved ELR (iELR). In the iELR method, when a mobile phone enters a cell to find that the cell counter value is less than the main counter value, or when a mobile phone enters a cell to register its location, it updates the main counter and the cell counter values as much as possible to reduce the future need for registrations. We show that our proposed iELR method provides better performance than the ELR method.

Keywords: Location registration, location-based registration, enhanced location-based registration, random walk, mobility model.

Manuscript received Feb. 23, 2015; revised July 26, 2015; accepted Aug. 10, 2015.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2012R1A1A2006725).

Ki Ho Seo (dandelian01@hanmail.net) is with the Department of Industrial & Systems Engineering, KAIST, Daejeon, Rep. of Korea.

Jang Hyun Baek (corresponding author, jbaek@jbnu.ac.kr) is with the Department of Industrial & Information Systems Engineering, Chonbuk National University, Jeonju, Rep. of Korea.

I. Introduction

With the advent of 4G networks, mobile subscribers have been increasing in number and more accelerated growth for high-quality services is expected [1]–[2]. In mobile cellular networks, location registration is the process by which a mobile registers its location and status to a network so that the network is able to efficiently page the mobile when an incoming call for the mobile arrives. Without registration, the network cannot know where the mobile is located and may have to page in many cells of the network [3]–[7].

It has been pointed out that a movement-based registration (MBR) method may be the most practical for location registration, since it is effective and easily implemented in the mobile and in the current mobile cellular network [6]–[9]; many studies on the MBR method have been performed [6]–[15]. In the MBR method, each mobile has a counter, whose value is compared with a movement threshold, M , to perform a location registration. A mobile registers its location whenever the number of cells it enters reaches the threshold M . In the MBR method, however, a mobile increases its counter value even when it reenters a cell already visited, which causes frequent location registrations compared to other location registration methods [6]–[7].

To solve this problem, a location-based registration (LR) method [9] and an enhanced LR (ELR) method [10] were proposed. For the present study, we consider the ELR method, which is superior to the LR method. Even though the number of location registrations of the ELR method is less than or equal to that of the LR method, there is still a lot of room for performance improvement in the ELR method.

We now propose an improved ELR (iELR) method. The performance of the ELR method can be improved considerably

if whenever a mobile enters a cell it updates the main counter and the cell counter values as much as possible for fewer future registrations. We have shown through various numerical results that our proposed iELR method provides better performance than the ELR method.

II. LR and ELR Methods

In this section, we briefly describe the LR and ELR methods, and explain the basic ideas for decreasing the location registration cost of the ELR method.

1. LR Method

In the LR method, each mobile maintains a counter, C , and a list of tuples, (I_i, C_i) , where I_i is the identifier (ID) of cell i , and C_i is the value of the counter immediately after the mobile enters cell i . The cells within the list are arranged in increasing order of C_i .

When a mobile enters a cell that is already in the list, its counter value is newly assigned with the value C_i , and all cells following cell i are removed from the list. The LR method always outperforms the MBR method [9]. An example is shown in Fig. 1.

<The LR method>

When the mobile enters cell i , the rules to store cell i in the list and update the counter are as follows:

- 1) If cell i is not in the list, then counter C is increased by one.
 - (a) If C is equal to the movement threshold M , then a location registration is triggered and the list is initiated with cell i .
 - (b) Otherwise, $C_i = C$, and pair (I_i, C_i) is added to the list.
- 2) If cell i is already in the list, then $C = C_i$, and all cells following cell i in the list are removed from the list.

Note that the memory size required in this method depends on the value of M . Because the maximum required size of the list is M , the memory size to store the list in the mobile is M times the size of a tuple.

2. ELR Method

The performance of the LR method can be enhanced considerably if, even when the mobile entering a cell finds that the cell is already in the list and then updates its main counter, the mobile does not remove any cells from the list as far as memory space permits.

In the ELR method, each mobile has a main counter C and a list of (I_i, C_i) , where I_i is the ID of cell i and C_i is the cell counter value of cell i , which is defined as the value of the main counter immediately after the mobile's most recent visit to cell i . It is

assumed that the mobile stores the cells to the list in the visiting order of the mobile. A flow chart is shown in Fig. 2.

<The ELR method>

When the mobile visits cell i , it decides whether it updates the list and the main counter as follows:

- 1) If cell i is not in the list, then the main counter C is increased by one, and the following cases are considered:
 - (a) If $C = M$, then the mobile registers its location. The main counter is reset ($C = 0$), and the list is also reset only with cell i .
 - (b) If $C < M$, then the following two cases are considered:
 - If the memory has a vacant space, then $C_i = C$, and the mobile adds (I_i, C_i) to the list.
 - Otherwise, then $C_i = C$. The mobile deletes the oldest cell j in the list, and adds (I_i, C_i) to the list.
- 2) If cell i is already in the list, then the following cases are considered:
 - (a) If $C_i < C$, then $C = C_i$.
 - (b) If $C_i \geq C$, then counter C is increased by one, and the following cases are considered:
 - If $C = M$, then the mobile registers its location. The main counter is reset ($C = 0$), and the list is also reset only with cell i .
 - If $C < M$, then $C_i = C$, and the mobile updates pair (I_i, C_i) in the list.

Note that, as shown in 2) (a), the mobile does not delete any cells in the list so long as memory space allows, which differs from the LR method. Also note that, as described in 2) (b), $C_i \geq C$ is possible in the ELR method because the memory may store more than M cells. Also note that after a mobile registers its location, its list is initiated. In other words, in the ELR method, when a location registration is performed, all cells except the current cell are deleted in the list.

3. Motivation

It is desirable to keep each mobile's list and main counter updated as much as possible. In the ELR method, even though the number of location registrations of the ELR method is always less than or equal to that of the LR method, there is a lot of room for performance improvement.

The performance of the ELR method can be improved considerably, when a mobile reenters a cell whose cell counter value is less than the main counter value, or when a mobile enters a cell to register its location, if it has enough memory space to keep previously visited cells and then updates the main counter and the cell counter values of its list as much as possible.

A. Case 1: When Mobile Enters Cell to Register its Location

In the ELR method, when a mobile registers its location, it

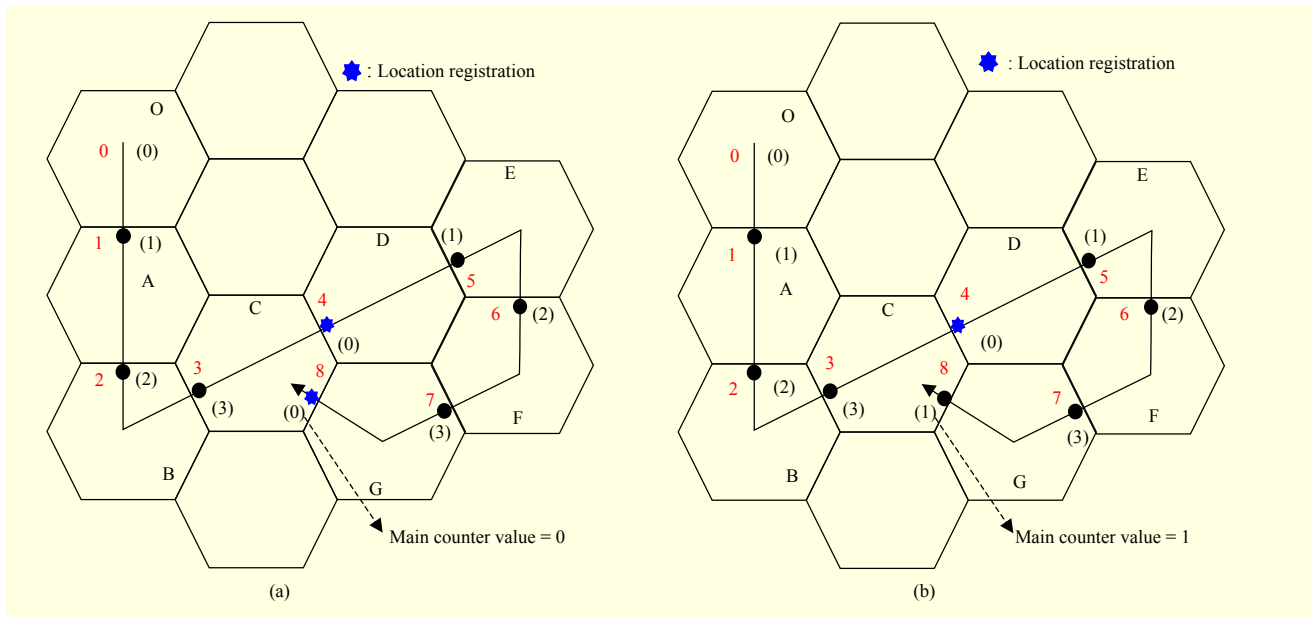


Fig. 1. Example path of mobile ($M = 4$): (a) ELR method and (b) iELR method.

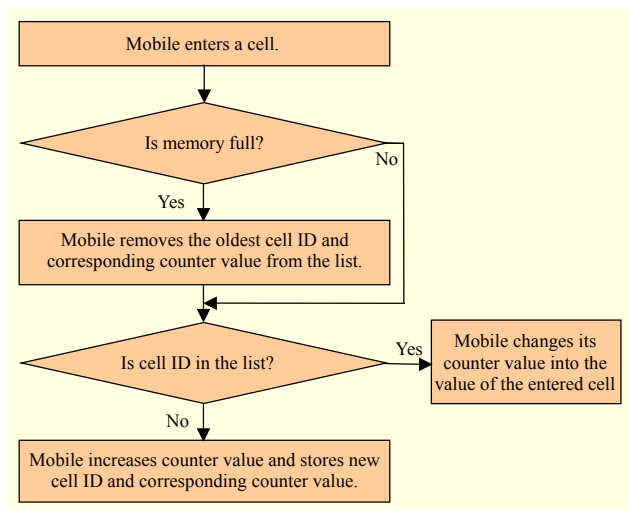


Fig. 2. Flow chart of ELR method.

resets its list. However, when a mobile registers its location, if it does not reset the list but keeps previously visited cells and then updates the list's cell counters values as much as possible, then it is possible to ultimately reduce the number of location registrations of the ELR method.

For example, in Fig. 1, consider the path where the mobile enters cell D (the 4th cell entrance) through cells B and C. In this case, if the ELR method is adopted, then the mobile entering cell D registers its location and resets the list to only have cell D ($C_D = 0$). However, note that when the mobile registers its location in cell D, if it does not reset the list but keeps previously visited cells and then updates the cell counter ($C_C = 1$, $C_B = 2$, ...) as much as possible (refer to No. 4 of the

iELR method in Table 1 and No. 4 of the iELR method in Table 2), then it is possible from time to time to reduce the main counter value of the mobile and to ultimately reduce the number of location registrations of the ELR method.

For example, in Fig. 1, the main counter value of the mobile reentering cell C (the 8th cell entrance) becomes 1. As a result, it does not perform the location registration that would be performed with the ELR method (refer to No. 8 of the iELR method in Table 1 and No. 8 of the iELR method in Table 2).

B. Case 2: When Mobile Reenters Cell Whose Cell Counter Value is Less Than Main Counter Value

In addition, when a mobile reenters a cell whose cell counter value is less than the main counter value, it can sequentially update the cell counter values of other cells in the list. Note especially that if a mobile reenters cells whose cell counters are updated, then it is possible for the mobile to reduce its main counter value.

For example, in Fig. 1, consider the case that the mobile reenters cell C (the 8th cell entrance) through cells F and G. In the ELR method, the mobile does not delete cell F and cell G in the list, but it keeps them and their cell counter values ($C_F = 2$ and $C_G = 3$). Even in this case, the mobile can update the cell counter of cell G with $C_G = 2$, since it knows that $C_C = 1$ and cell G is the neighboring cell of cell C (refer to the 8th cell entrance in Fig. 1(b) and No. 8 of the iELR method in Table 1). Note that if the mobile reenters cell G, then it is possible for the mobile to reduce its main counter value. As a result, it ultimately decreases the number of location registrations.

Overall, when the mobile enters a cell then by updating the

Table 1. Lists and variables of two methods for Fig. 1.

No.	ELR scheme								iELR (3M) scheme												
	Tuples				C	L	N	S	Tuples								C	L	N	S	
1	O	A			1	1	1	0	O	A								1	1	1	0
	0	1							0	1											
2	O	A	B		2	2	2	0	O	A	B						2	2	2	0	
	0	1	2						0	1	2										
3	O	A	B	C	3	3	3	0	O	A	B	C					3	3	3	0	
	0	1	2	3					0	1	2	3									
4	D				0	4	4	0	O	A	B	C	D				0	4	4	0	
	0								4	3	2	1	0								
5	D	E			1	5	5	0	O	A	B	C	D	E			1	5	5	0	
	0	1							4	3	2	1	0	1							
6	D	E	F		2	6	6	0	O	A	B	C	D	E	F		2	6	6	0	
	0	1	2	3					4	3	2	1	0	1	2						
7	D	E	F	G	3	7	7	0	O	A	B	C	D	E	F	G	3	7	7	0	
	0	1	2	3					4	3	2	1	0	1	2	3					
8	C				0	8	8	0	O	A	B	C	D	E	F	G	C	1	7	8	1
	0								4	3	2	1	0	1	2	2	1				

main counter and the cell counter values of other cells in the list as much as possible, it is possible for the mobile to reduce its main counter value and thereby ultimately reduce the number of location registrations.

In summary, the performance of the ELR method can be improved considerably, especially when a mobile reenters a cell whose cell counter value is less than the main counter value, or when a mobile enters a cell to register its location, if the mobile has enough memory space to keep previously visited cells listed and updates the cell counter values of the list and main counter as much as possible.

Henceforth, our improved ELR method is called the iELR method. Note that the iELR method uses the same memory space as the ELR method. Furthermore, considering the very low price of memory in the market, it is not economically difficult to extend memory space in both methods.

It is obvious that the iELR method needs more processing to update the cell counters compared to the ELR method. However, if we use Dijkstra's algorithms, then it is possible to update the cell counters with $O(M^2)$ running time on a mobile having cM ($c = 2, 3$) cells in the list, as shown in Section III.

III. iELR Method

In the iELR method, in a similar manner to the ELR method,

each mobile has a main counter and a list of tuples (I_i, C_i) . The ID of cell i is denoted by I_i , and C_i is the cell counter value of cell i , which is defined as the value of the main counter right after the mobile enters cell i . In general, since the maximum size of the list is greater than M , the rules for removal of cells and for updating cell counter values of kept cells in the list are different from those of the ELR method. The mobile stores the cells to the list in the visiting order of the mobile; that is, the order of the cells in the list signifies the mobile's visiting order for the cells. The iELR method operates as follows (a flow chart of its operations is shown in Fig. 3):

<The iELR method>

When the mobile enters cell i , it decides whether to update the list and the main counter as follows:

- 1) If cell i is not in the list, then the main counter, C , is increased by one, and the following cases are considered:
 - (a) If $C = M$, then the mobile registers its location ($C = C_i = 0$), adds pair (I_i, C_i) to the list, and updates the list by the rule of update.
 - (b) If $C < M$, then $C_i = C$, and the mobile adds pair (I_i, C_i) to the list.
- 2) If cell i is in the list, then the following cases are considered:
 - (a) If $C_i < C$, then $C = C_i$, and the mobile adds pair (I_i, C_i) to the list and updates the list by the rule of update.
 - (b) If $C_i \geq C$, then the main counter, C , is increased by one,

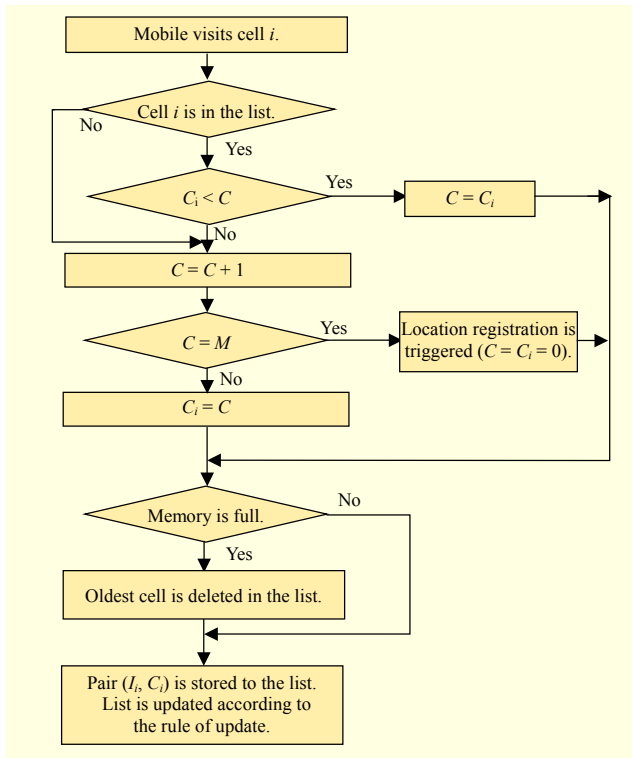


Fig. 3. Flow chart of iELR method.

and the following cases are considered:

- (i) If $C = M$, then the mobile registers its location ($C = C_i = 0$), adds pair (I_i, C_i) to the list, and updates the list by the rule of update.
- (ii) If $C < M$, then $C_i = C$, and the mobile adds pair (I_i, C_i) to the list and updates the list by the rule of update.

Note that, as is shown in 1)(a) and 1)(b), a check must be made to see whether the memory has vacant space before adding (I_i, C_i) to the list. If the memory has no vacant space, then the mobile should delete the oldest cell in the list to secure vacant space for (I_i, C_i) .

Also note that, as is shown in 1)(a) and 2)(b)(i), even when the mobile registers its location, it does not delete any cells in the list but updates all cells in the list by the rule of update, which differs from the ELR method.

Finally note that, as shown in 1)(a), 2)(a), and 2)(b), every time a mobile enters a cell, it updates the list as much as possible, which is significantly different from the ELR method.

<The rule of update >

When a mobile enters a cell, it should update the cell counter values in the list as follows:

- 1) When a mobile enters a cell to register its location (in other words, $C = M$), Algorithm 1 of Section III-3 is applied to update the cell counter values of the list.
- 2) When a mobile reenters a cell whose cell counter value is

less than the main counter value (in other words, $C < M$), Algorithm 2 of Section III-3 is applied to update the cell counter values in the list.

1. Example (Changes of Counter Values in Fig. 1)

Now, let us consider Fig. 1 again. Table 1 shows the changes of the main counter values and the cell counter values of the list for Fig. 1 in the iELR method. An iELR method that can store a maximum of n cells is signified by iELR(n).

In Table 1, the following notations are used for illustrative purposes:

- C : value of the main counter
- N : the number of new cells visited by the mobile that are not in the list between two consecutive incoming calls
- S : the cumulative reduction of the main counter value between two consecutive incoming calls
- L : $N - S$

From Table 1, we can see that the iELR method provides fewer registrations compared to the ELR method.

As shown in the flow chart of the iELR method, whenever a mobile enters a cell, it determines its main counter value and then updates the cell counter values of other cells in the list. Referring to Table 1, we now explain how to update the cell counter values in the list.

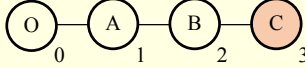
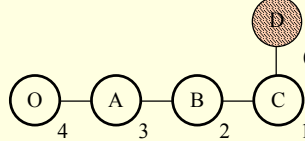
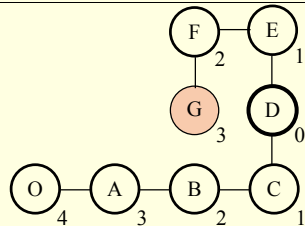
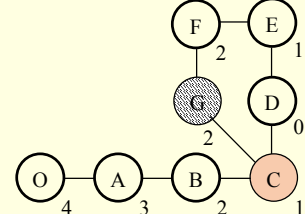
2. Algorithm for Updating Cell Counter Values in List

In the iELR method, for the purpose of updating cell counter values in the list, a mobile maintains a graph in which nodes denote cells in the list and links denote that two nodes connected by a link are neighboring cells. Whenever a mobile enters cell i , it modifies the graph in which node i is connected to the last node by a link. The distance on every link is set to a value of one, since two nodes connected by a link are neighboring cells.

In this study, let cell j that has the least cell counter value ($C_j = s$) in the graph be a source to obtain the shortest distance from a source to each cell. Then, the final updated cell counter of each cell can be obtained by adding the cell counter value of the source to the shortest distance from the source to each cell. Note that, if the graph has node k at which a mobile has most recently registered its location ($C_k = 0$), then the shortest distance from node k to each node is the updated cell counter value of each cell.

Table 2 shows the changes in the graphs for Fig. 1 with the iELR method. In the graph of Table 2, a shaded (hatched) node indicates that the node has been updated by the rule of update just after the mobile enters the current cell to determine its main counter.

Table 2. Graphs derived from Fig. 1 for iELR (3M) scheme.

No.	Graph	iELR (3M) scheme										C
		Tuples										
...
3		O	A	B	C							3
		0	1	2	3							
4		O	A	B	C	D						0
		4	3	2	1	0						
...
7		O	A	B	C	D	E	F	G			3
		4	3	2	1	0	1	2	3			
8		O	A	B	C	D	E	F	G	C		1
		4	3	2	1	0	1	2	2	1		

○: Source node ●: Update node ◐: Current node

In this study, Dijkstra's algorithm [16] is applied to obtain the shortest distance from a source to each node, whose running time is known to be $O(n^2)$ on a graph having n nodes. So, when a mobile enters a cell, the running time for updating the cell counters in the list is $O(M^2)$ on a mobile having cM ($c=2, 3$) cells in the list. However, in the iELR method, when a mobile enters a cell, it is not necessary to calculate all of the shortest distances from a source to each node again, since only one node and one link is added into the graph.

In the iELR method, the mobile updates the cell counter values of cells connected to cell i first when either (a) a mobile reenters cell i whose cell counter value is less than the main counter value (for example, No. 8 in Table 2) or (b) it enters cell i to register its location (for example, no. 4 in Table 2). Among them, if there are cells whose cell counter values are updated, then the mobile next updates the cell counters of cells connected to those updated cells. Similarly, by performing this updating process until there are no updates left, all of the updatable cell counters in the list can be updated.

3. Pseudocodes for Two Types of Modified Dijkstra's Algorithm to Update Cell Counter Values

In the iELR method, the mobile should update the cell counter values in the list in either of the following cases:

- 1) when it enters cell i to register its location (for example, No. 4 in Table 2), or
 - 2) when a mobile reenters cell i whose cell counter value is less than the main counter value (for example, No. 8 in Table 2).
- Depending on which case the mobile meets, two different modified versions of Dijkstra's algorithm are applied to update the cell counter values in the list.

In the case of 1) above, Algorithm 1 is applied to update the cell counter values in the list. In this case, the cell counter value of every cell in the list must be checked to be updated.

Algorithm 1. Modified Dijkstra's algorithm for a mobile that enters a cell to register its location.

```

1 function Dijkstra_iELR_register(Graph, source):
2   dist[source] := 0 // Distance to source
3   for each node v in Graph: // Initialize

```

```

4  if  $v \neq source$ 
5     $dist[v] := infinity$  // Unknown distance to  $v$ 
6  end if
7  add  $v$  to  $Q$  // All nodes in the node set  $Q$  (unvisited nodes)
8  end for
9  while  $Q$  is not empty: // Find the shortest path
10    $u := node$  in  $Q$  with min  $dist[u]$  //  $u$  has the least distance in  $Q$ 
11   delete  $u$  from  $Q$ 
12   for each neighbor  $v$  of  $u$ : // Search for each neighbor  $v$  of  $u$ 
13      $alt := dist[u] + 1$  // Distance to  $v$  if it goes through  $u$ 
14     if  $alt < dist[v]$ : // A shorter path to  $v$  is found
15        $dist[v] := alt$  // Distance to  $v$  is replaced
16     end if
17   end for
18 end while
19 return  $dist[ ]$ 
20 end function

```

In Algorithm 1, the code “ $u := node$ in Q with min $dist[u]$ ” (line 10) searches for a node “ u ” that has the least distance in the node set “ Q .” In the code “ $alt := dist[u] + 1$ ” (line 13), “1” is the distance between two neighboring nodes. The variable “ alt ” (line 14) is the distance from the source node to the node v , if it goes through u . If this path is shorter than the current path, then this “ alt ” path replaces the current shortest path.

In the case of 2) above, Algorithm 2 is applied to update the cell counter values in the list. In this case, only those cells whose cell counter values are greater than d may be updated, where d is the cell counter value of the reentered cell i .

Algorithm 2. Modified Dijkstra’s algorithm for the mobile that enters a cell in the list.

```

1 function Dijkstra_iELR_reenter(Graph, reenternode):
2    $d := dist[reenternode]$  // Distance to reenternode
3   for each node  $v$  in Graph: // Initialize
4     if  $dist[v] > d$ :
5       add  $v$  to  $Q$  // Updatable nodes in the node set  $Q$ 
6     end if
7   end for
8   add reenternode to  $Q$ 
9   Same procedure of line 9–19 of Algorithm 1
10 end function

```

In Algorithm 2, the code “ $dist[v] > d$.” (line 4) searches for a node v that has a distance larger than d since its distance may be updated if it goes through “reenternode.”

It is an interesting topic to improve the algorithm for updating the cell counter values, but we retain this for further study since it is not the core aim of this paper.

IV. Performance of iELR Method

In this section, it will be shown that the location registration

cost of the iELR method is less than or equal to that of the ELR method. We obtained and compared the location registration costs of two methods on radio channels. It is assumed that the processing cost for updating cell counter values is so low as to be ignored.

Let us introduce several random variables again to obtain an analytical expression for the location registration cost and to compare the location registration costs of two methods. These are as follows:

- K : the number of visited cells between two consecutive incoming calls
- N_X : the number of new cells visited by the mobile that are not in the list of the X method between two consecutive incoming calls
- S_X : the cumulative reduction of the main counter value of the X method between two consecutive incoming calls
- L_X : $N_X - S_X$

Then, the following property can be obtained.

Property. $N_{ELR} \geq N_{iELR}$, $S_{ELR} \leq S_{iELR}$, and $L_{ELR} \geq L_{iELR}$ for the cells visited by the mobile between two consecutive incoming calls.

Let us consider these random variables for Fig. 1. Note again that Table 1 shows the values of random variables for Fig. 1. We can then obtain the location registration costs between two consecutive incoming calls as follows:

$$C_U^{ELR} = U \sum_{i=1}^{\infty} i \sum_{k=iM}^{\infty} \alpha(k) \sum_{l=iM}^{(i+1)M-1} \Pr[L_{ELR} = l | K = k],$$

$$C_U^{iELR} = U \sum_{i=1}^{\infty} i \sum_{k=iM}^{\infty} \alpha(k) \sum_{l=iM}^{(i+1)M-1} \Pr[L_{iELR} = l | K = k].$$

In the above, U is the unit cost for one location registration and $\alpha(k)$ is the probability that a mobile crosses k cell boundaries between two consecutive incoming calls [6]–[10]. To get the value of $\Pr[L | K]$ in the above equations, we used computer simulations since this calculation is so complicated. As a final result for the performance comparison, the following corollary can be obtained.

Corollary. For the cells visited by the mobile between two consecutive incoming calls, the number of location registrations of the iELR method is less than or equal to that of the ELR method. In other words,

$$C_U^{iELR} = U \sum_{i=1}^{\infty} i \sum_{k=iM}^{\infty} \alpha(k) \sum_{l=iM}^{(i+1)M-1} \Pr[L_{iELR} = l | K = k]$$

$$\leq U \sum_{i=1}^{\infty} i \sum_{k=iM}^{\infty} \alpha(k) \sum_{l=iM}^{(i+1)M-1} \Pr[L_{ELR} = l | K = k] = C_U^{ELR}.$$

Finally, letting C_p be the paging cost between two consecutive incoming calls, we can obtain the total signaling cost between two consecutive incoming calls, C_T , for location registration and paging as in the following:

$$C_T = C_U + C_P = C_U + V[1 + 3M(M - 1)].$$

In the above, V is the unit cost for paging one cell, and $[1 + 3M(M - 1)]$ is the total number of hexagonal cells in a location area. Also note that, in this study, simultaneous paging is assumed when an incoming call arrives [8], [10], [12].

V. Numerical Results

In this study, we assume a random walk mobility model in a hexagonal cell configuration [6]–[9]. We also assume that the incoming call arrivals and the cell residence time follow a Poisson process with λ_c and an exponential distribution with λ_m , respectively. We obtained the numerical results, assuming the following environments [6]–[7]: $\lambda_c = 1$, $\lambda_m = 4$, $U = 10$, $V = 1$.

Figure 4 shows the location registration costs of the two methods between two consecutive incoming calls for various thresholds, assuming that the call-to-mobility ratio (CMR) is 1/4. As the CMR is defined as λ_c/λ_m , less CMR means higher mobility of the mobile. For example, $CMR = 1/4$ means that a mobile enters four cells between two consecutive incoming calls. From the figure, it can be seen that the registration cost of the iELR method is always less than or equal to that of the ELR method. It can also be observed that iELR(3M) has a lower location registration cost than iELR(2M), but the difference is small.

Figure 5 shows the location registration cost of the iELR(2M) method for various CMRs. We consider three cases of $CMR = 1/2$, $1/4$, and $1/10$. The location registration cost for $CMR = 1/10$ can be seen in Fig. 6. From Figs. 5 and 6, it can be seen that, regardless of the CMR values, the iELR method shows a significant cost reduction compared to the ELR method. It can also be observed that, as the threshold increases, the location registration cost decreases but the reduction ratio (ELR-iELR)/ELR increases.

Figure 6 shows the total signaling cost of the two methods for various thresholds, given $CMR = 1/10$. From Fig. 6, it can be seen that both of the methods have the least total signaling cost when threshold $M = 3$, and that the total signaling cost of the iELR method is reduced by about 4.0% (and registration cost is reduced by about 7.6%) compared to the ELR method. It can also be observed that for the optimal threshold, $M^* = 3$, the total signaling cost of the iELR method is reduced by about 6.4% (and registration cost is reduced by about 12.0%) compared to the original LR method. A more distinct cost reduction is possible in the iELR method, as the CMR decreases, since a low CMR means high mobility and leads to many registrations.

In conclusion, in view of the various numerical results, our proposed iELR method provides better performance compared

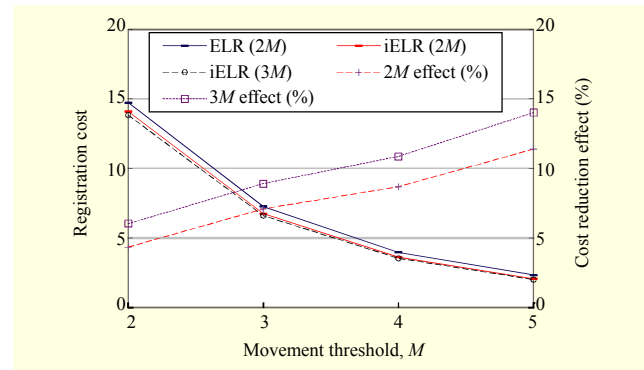


Fig. 4. Location registration costs of two methods ($CMR = 1/4$).

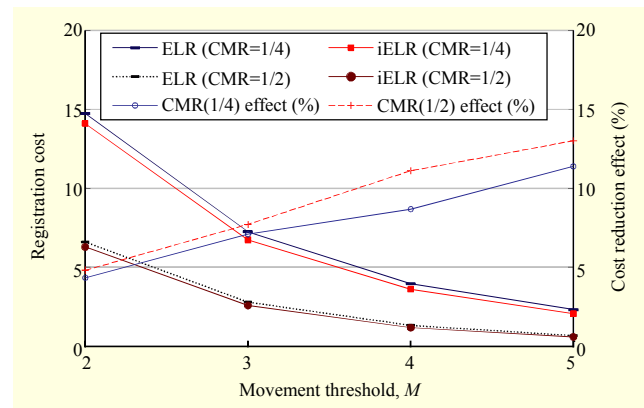


Fig. 5. Location registration costs for various CMRs.

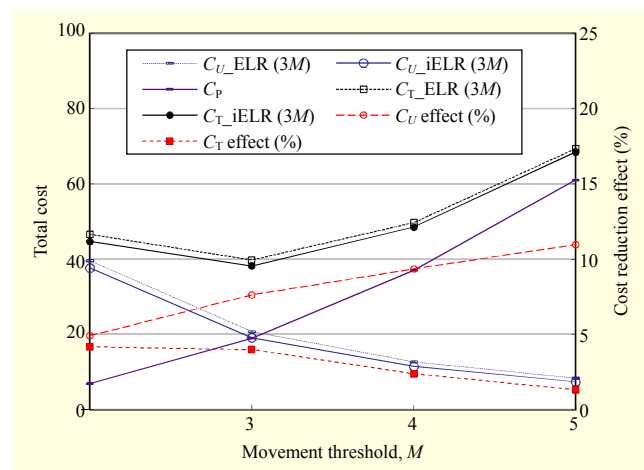


Fig. 6. Total signaling costs for two methods ($CMR = 1/10$).

to the ELR method in every case.

VI. Conclusion

In our study, we considered an enhanced location-based location registration (ELR) method [10]. We proposed an improved version of the ELR method with which a mobile

does not delete previously visited cells from the list but keeps them in memory as space permits. And whenever a mobile enters a cell, it updates the main counter and the cell counter values of its list as much as possible so as to minimize the number of future registrations. The performances of two methods were analyzed and compared through computer simulations by a random walk mobility model in a hexagonal cell configuration. The results of the simulations for various circumstances showed that our proposed iELR method is always superior to the ELR method. These results can be used to select and operate an appropriate registration method based on network conditions.

References

- [1] Y.B. Lin et al., "Performance Evaluation of LTE eSRVCC with Limited Access Transfers," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, May 2014, pp. 2402–2411.
- [2] R.H. Liou et al., "Performance of CS Fallback for Long Term Evolution Mobile Network," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, Oct. 2014, pp. 3977–3984.
- [3] TIA/EIA/IS-95-B, *MS-BS Compat. Standard for Dual-Mode Wideband Spread Spectrum Cellular System*, 1999, pp. 6-155–6-169.
- [4] J.H. Baek, T.H. Lee, and C.S. Kim, "Performance Analysis of 2-Location Distance-Based Registration in Mobile Communication Networks," *IEICE Trans. Commun.*, vol. E96-B, no. 3, Mar. 2013, pp. 914–917.
- [5] R.H. Liou, Y.B. Lin, and S.C. Tsai, "An Investigation on LTE Mobility Management," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, Jan. 2013, pp. 166–176.
- [6] I.F. Akyildiz, J.S.M. Ho, and Y.B. Lin, "Movement-Based Location Update and Selective Paging for PCS Networks," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, Aug. 1996, pp. 629–638.
- [7] J.H. Baek and B.H. Ryu, "An Improved Movement-Based Registration in Personal Communication System Networks," *IEICE Trans. Commun.*, vol. E83-B, no. 7, July 2000, pp. 1509–1516.
- [8] J. Li, H. Kameda, and J. Li, "Optimal Dynamic Mobility Management for PCS Networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, June 2000, pp. 319–327.
- [9] Z. Mao and C. Douligieris, "A Location-Based Mobility Tracking Scheme for PCS Networks," *Comput. Commun.*, vol. 23, no. 18, Dec. 2000, pp. 1729–1739.
- [10] J.H. Baek et al., "An Enhanced Location-Based Location Update Scheme in Mobile Cellular Networks," *ETRI J.*, vol. 27, no. 4, Aug. 2005, pp. 457–460.
- [11] Y. Zhu and V.C.M. Leung, "Optimization of Sequential Paging in Movement-Based Location Management Based on Movement Statistics," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, Mar. 2007, pp. 955–964.
- [12] X. Wang et al., "Cost Analysis of Movement-Based Location Management in PCS Networks: An Embedded Markov Chain Approach," *IEEE Trans. Veh. Technol.*, vol. 63, no. 4, May 2014, pp. 1886–1902.
- [13] Y.W. Chung, J.K. Kwon, and S. Park, "Modeling and Performance Analysis of an Improved Movement-Based Location Management Scheme for Packet-Switched Mobile Communication Systems," *Scientific World J.*, vol. 2014, Mar. 2014.
- [14] R.M. Rodriguez-Dagnino and H. Takagi, "Application of Renewal Theory to Call Handover Counting and Dynamic Location Management in Cellular Mobile Networks," *European J. Operational Res.*, vol. 204, no. 1, July 2010, pp. 1–13.
- [15] K. Li, "Cost Analysis and Minimization of Movement-Based Location Management Schemes in Wireless Communication Networks: A Renewal Process Approach," *Wireless Netw.*, vol. 17, no. 4, May 2011, pp. 1031–1053.
- [16] H.A. Taha, *Operations Research: An Introduction*, Upper Saddle River, NJ, USA: Prentice Hall, 2007, pp. 243–262.



Ki Ho Seo received his BS degree in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2011. He is currently in a combined MS and PhD program at KAIST. His research interests include operations research and its application to communication networks.



Jang Hyun Baek received his BS, MS, and PhD degrees in industrial engineering from Seoul National University, Rep. of Korea, in 1986, 1988, and 1997, respectively. He joined the Switching Technology Division of ETRI in 1989 and worked later in the Mobile Communication Division. In 1998, he joined the faculty of the College of Engineering, Chonbuk National University, Jeonju, Rep. of Korea, where he is currently a full-time professor with the Department of Industrial and Information Systems Engineering. His current research interests include stochastic processes and optimization theory and their applications to design, modeling, and performance analysis of communication networks.