Available online at www.sciencedirect.com
ScienceDirect
J. Parallel Distrib. Comput. 67 (2007) 430–447
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

Clustering techniques for dynamic location management in mobile computing

Javid Taheri, Albert Y. Zomaya*

School of Information Technology, The University of Sydney, Sydney, NSW 2006, Australia

Received 24 March 2006; received in revised form 16 December 2006; accepted 29 December 2006
Available online 19 January 2007

Abstract

This paper presents a novel approach based on clustering algorithms in combination with the location area (LA) scheme to solve the mobility management problem. Users' movement history is used by the network to predict future paging decisions. This approach integrates the LAs scheme and efficient clustering algorithms to find a network topology which can lead to massive savings in the number of signals made to locate users in the network. The approach is tested with several networks to show its advantages to the current GSM standards. The results provide new insights into the mobility management problem.

Keywords: Mobile location management; GSM network; Data clustering; Dynamic location management

1. Introduction

Numerous companies and service providers are pursuing a fully integrated service solution for wireless mobile networks. Current voice, fax, and paging services will be combined with data transfer, video conferences, and other mobile multimedia services to make the next generation of wireless mobile networks more attractive. These networks will support a true combination of both real-time and non-real-time services to form a global personal communication network [29,1]. In order to support such a wide range of data transfer and user applications, mobility management has to be considered when designing infrastructure for wireless mobile networks.

Mobility management involves two processes: location management and handoff management. Location management enables the wireless network to discover the current point of attachment of a mobile terminal and deliver calls to it, while handoff management enables the mobile network to locate roaming mobile terminals for call delivery and to maintain connection as the mobile terminal is moving around. During the first stage of location management, known as location registering or location update, the mobile terminal periodically informs the network its new access point and helps the network to authenticate the user and revise the users' location profiles. The second stage is call delivery, in which the wireless mobile network is queried for a mobile terminal location and the current position of that terminal is found [29,1].

On the other hand, handoff primarily represents a process of changing some of the parameters of a channel (frequency, time slot, spreading code, or a combination of these) while the current connection is in use [6,57]. The handoff process usually consists of two phases: handoff initialization and handoff enabling phase. In the handoff initialization phase, the quality of the current communication channel is considered to decide when the handoff process should be triggered. In the handoff enabling phase, the allocation of new resources by new base station is initiated and processed. Poorly designed handoff schemes tend to generate huge signaling traffic, and thereby a dramatic decrease in quality of service (QoS) of integrated services in the wireless network.

Mobility management requests are often initiated either by a mobile terminal movement (crossing a cell boundary) or by deterioration in the quality of a signal received on a currently allocated channel. Due to the anticipated increase in the use of wireless services in the future, the next generation of mobile networks should be able to support a huge number of users

* Corresponding author. Fax: +61 2 9351 3838.
E-mail addresses: javidt@it.usyd.edu.au (J. Taheri), zomaya@it.usyd.edu.au (A.Y. Zomaya).

0743-7315/$ - see front matter Crown Copyright © 2007 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2006.12.005

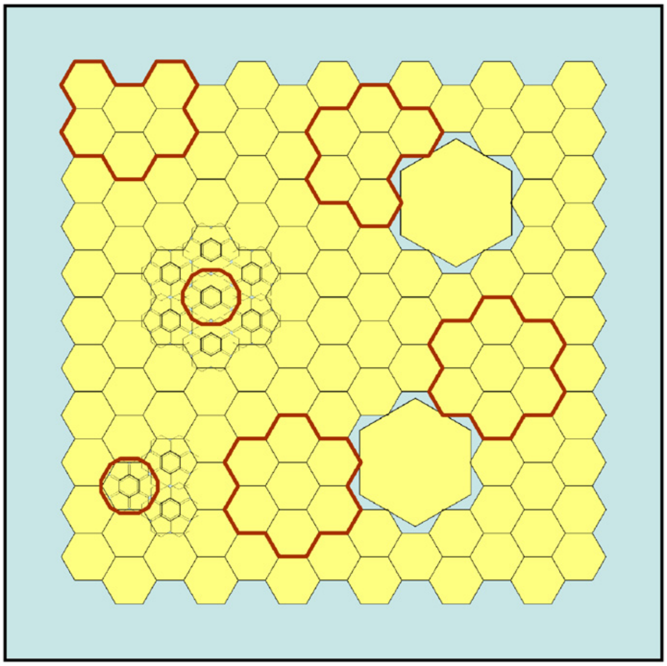

Fig. 1. An example of a GSM network.

and their bandwidth requirements. Furthermore, more frequent handoffs will occur when the size of the cells becomes smaller or there is a drastic change in the propagation condition of a signal. Therefore, mobility management becomes more crucial in the next generation of mobile networks.

Fig. 1 shows an example of a GSM network [29,1,6,57,5,38,4, 54]. In this network, cells are grouped together into regions. Each region contains the whole allotted frequency spectrum, while each cell of the group uses a part of the allocated frequency. The same frequency can be used in other regions by carefully considering the minimum distance between cells to avoid cross talking [5,38,4,54]. On the other hand, as the demand for wireless services increase, the size of the cells becomes smaller and the reusability of the allocated frequencies becomes more serious. As a result, network management, and consequently, location management turns into a serious problem, and, efficient techniques will be needed to ensure delivery of all incoming calls even in the tiniest of cells.

This paper presents an approach that employs two different clustering techniques (statistical and k-means) to solve the dynamic mobility management problem. Section 2 provides a general overview of different location management strategies. In Section 3, a general description is provided of how the cost is calculated for the location areas (LAs) scheme followed by an overview of the simulation framework in Section 4. Sections 5 and 6 go over the clustering techniques used in this work and the different classes of location management strategies, respectively. Sections 7 and 8 detail the proposed approaches along with extensive simulations. Finally, discussions and conclusions follow in Sections 9 and 10, respectively.

2. Location management strategies

Location management strategies can be categorized into two main groups: static and dynamic. In static schemes, the way that users are paged is very much the same for all users and is determined in an off-line manner and it does not take into account any particular information about the users [7,37,2,10,51,52,39,15,11,40–44,34], while in dynamic schemes, different network topologies are used for different users [3,8,9,12–14,16–23,25–28,30–33,35,36,46–48,50,53, 55,56]. These topologies are highly related to the movement pattern and calling behavior of each user. It is obvious that dynamic schemes are much more complex than static ones and require more computation capabilities in the network. Thus, the implementation of static schemes is more popular [39,15,11,40–44,34]. In general, a location update strategy must use nominal network resources to manage user tracking and should not require massive computations.

2.1. Basic location management strategies

This section briefly overviews some of the most common techniques employed for static location update [39,15,11, 40–44,34].

(1) *Always update strategy*: In this scheme, a mobile terminal updates its location whenever it crosses a cell boundary. This enables the network to locate the user in minimum time for each incoming call. This means that the network will page the user only in the last updated cell. This scheme generates a massive number of unnecessary location update signals.

(2) *Never update strategy*: A mobile terminal never updates its location, and consequently, for each incoming call the network must page the user in all cells of the network. It is also obvious that this strategy is highly inefficient, especially when there are massive numbers of cells in the network. Naturally, as the number of users in the system increases, the number of simultaneous paging signals would also increase.

(3) *Time based strategy*: In this method, a user updates its location after a predefined time span [7,37]. In the case of an incoming call, the network will page the user in the last update cell and the possible cells that the user might be in after the last location update. Although this strategy seems better than the previous two, some disadvantages remain [37]. For example, if the mobile terminal changes its location very frequently it becomes impossible to locate it in a timely fashion.

(4) *Movement based strategy*: The user performs a location update after passing a predefined number of cells, M. In this case, the network attempts to page the user in a radius of M cells starting from the last update cell [2,10]. The main disadvantage of this strategy is in the case of the users who periodically crisscross (zigzag) adjacent cells of a network.

(5) *Distance based strategy*: In this case, the user updates its location after traveling for a predefined distance, R, form the last update cell [51,52]. The network can page the user in all cells that are physically within radius R of the last updated location. The main problem in this method is the need for knowledge of the topography of the network. In this case, each

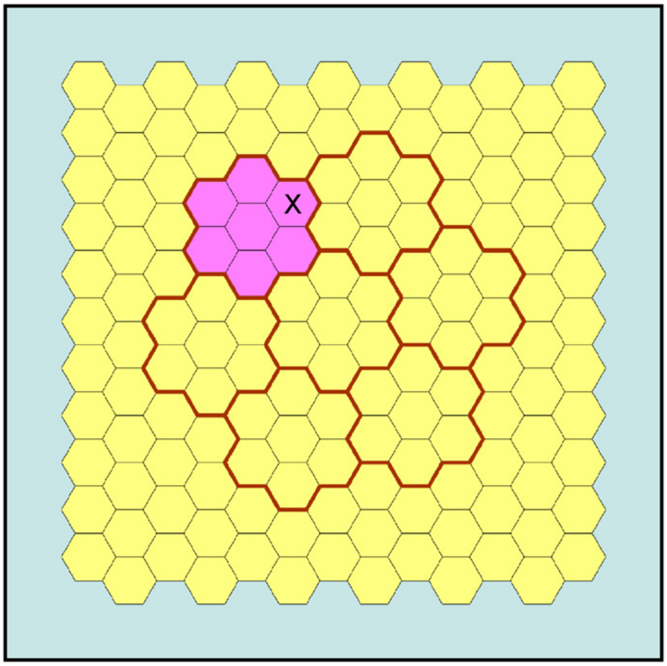

Fig. 2. Location areas configuration.

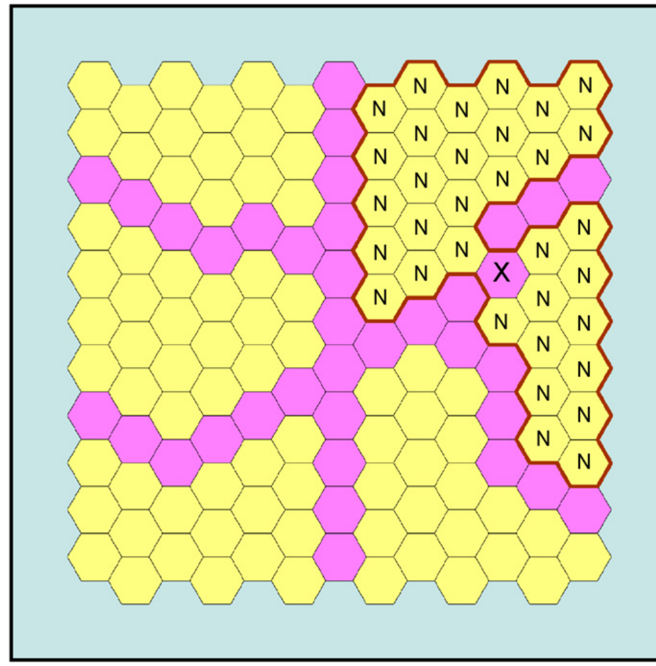

Fig. 3. Paging cells configuration.

mobile terminal should have complete information about network topology and must be able to determine its geographical location as well. This makes the design of mobile terminals more complicated and expensive than, for example, current designs.

(6) *LAs scheme*: This technique is adopted in the current GSM networks, where adjacent cells are grouped together to form a LA [39,15,42–44,30]. In this case, the user performs an update whenever it leaves a LA and moves into a new one. Therefore, in the case of an incoming call, the network must page the user in all the cells of the last update LA. Fig. 2 shows a typical GSM network layout. In this case, if the user updates its location in cell 'X', then for each incoming call, all cells of the current LA (shown in gray) will page the user.

(7) *Paging cells scheme*: In this case, the user makes an update whenever it passes through predefined cells known as *paging cells* [40,41,18]. A paging neighborhood is defined for each paging cell. The paging neighborhood of each paging cell contains all non-paging cells of the network that must page the user in the case of an incoming call. For example, if the paging cells are shown in gray in Fig. 3, then the paging neighbors of the paging cell marked as X are all cells marked as N. Usually, paging cells are assigned so that they split the entire network to several smaller networks. Then, the never update scheme is used in each of these sub-networks.

2.2. *Advanced location management strategies*

Traditional location management techniques for mobile networks are very conservative and there is a need for more 'intelligent' techniques' that endow networks with better location prediction capabilities. An intelligent strategy would normally have two components: (1) modeling users' behavior based on movement history, and (2) an intelligent algorithm that utilizes this information to make predictions on future movements. Several algorithms and strategies have been suggested to solve this problem. The use of Markov models is one of the most popular techniques in solving this problem [23,19,17,27,25]. Using history based [19], distance based [27,18], movement-based [26], time based strategies [35,36], and one-dimensional techniques [28] are some of the other popular techniques cited in the literature.

In some of the existing approaches for solving dynamic mobility management problems, users are divided into two main groups: a low mobility group and a high mobility group. The low mobility users never update their location explicitly, while the high mobility users update their location whenever they enter a new LA. As a consequence, low mobility users might miss some incoming calls. To overcome this problem, if the network cannot find a user in a given LA, it employs a 'ring growing' paging strategy, until it finds the mobile user [21,31]. However, the use of intelligent paging techniques can reduce the total paging cost of the network and this is seen as the major advantage for using dynamic location management strategies, even in the case of traditional GSM networks [46,47]. The use of techniques such as Fuzzy Logic is suggested as one reasonable approach to build more intelligent mobile networks in [20]. Using sequential paging inside the user's LA, which is known as paging time, is another approach employed to reduce the total cost of the network management proposed in [38]. In another method, a given city is divided into two main types of LAs. The mobile terminals update their location whenever they enter or leave the highway sections of the city, otherwise, they just update their location using a predetermined frequency of updating [8].

The use of statistical techniques to 'learn' the behavior of the users is another popular approach that can lead to better location management in this case [53]. These techniques are normally used to cluster each user's behavior patterns. Then, an 'extracted' time-location table is uploaded onto each mobile device and if the mobile terminal follows the movement patterns in the table, it does not need to update its location, otherwise, it has to [53]. However, all these techniques have some shortcomings whether through making unrealistic assumptions about the network or the movement patterns. This paper introduces a novel dynamic LAs scheme that employs a two-step paging algorithm that can provide an efficient solution to this problem with more realistic assumptions.

3. Location management cost in the LAs scheme

Many algorithms have been proposed to solve the location management problem [7,37,2,10,51,52,39,15,11,40–44]. However, there is a need to develop a framework that can be used to compare these techniques. A location management cost needs to be defined to evaluate each approach. The location management cost usually consists of two main parts: updating and paging costs. The updating cost is the portion of the total cost due to location updates performed by mobile terminals in the network while paging cost is caused by the network during a location inquiry when the network tries to locate a user. The total cost of location management also involves other parameters, such as, the cost of database management to register users' locations, the cost of the wired network (backbone) that connects the base stations to each other, the cost of switching between base stations in the case of handoff and call diverting, and several other components. However, these costs are assumed to be the same for all location management strategies in general. As a result, the combination of location update and paging costs are considered to be sufficient to compare the different approaches. Therefore, the total cost of a location management scheme is given as [40–44]

$$\mathit{Cost} = \beta \times N_{\mathrm{LU}} + N_{\mathrm{P}},$$

where N_{LU} is the total number of location updates, N_{P} is the total number of paging transactions, and β is a constant representing the cost ratio of a location update to a paging transaction in the network. The number of location updates is usually caused by the movement of the users in the network, while the number of the paging transactions is highly related to the number of incoming calls. Evidence shows that the cost of each location update is much higher than the cost of a paging transaction, because of a complex procedure that has to be executed every time a location update is performed [40–44]. On the other hand, most of the calls of a mobile user are the incoming calls. Therefore, if the user moves in the network without making any call, the network will undergo a huge number of useless transactions. Thus, in most cases the cost of a location update is considered to be 10 times more than that of a paging transaction, i.e. $\beta = 10$ [40–44].

Based on the LA scheme, a location update transaction takes place when a user changes its current LA. In this case, all entrances to LAs of the network should be added to compute this cost. On the other hand, the paging cost must be considered when a user has an incoming call. Thus, all cells must be considered in calculating this cost. To clarify this, assume an instance LA scheme (marked as gray) in the network in Fig. 4. In this configuration, the flows of the total number of users who enter the instance LA via its boundary cells are shown in Fig. 4a and are equal to

$$\begin{aligned}\text{Number of location updates} = {} & 2343 + 4323 + 4342 + 5435 \\ & +1101 + 4343 + 9832 + 1523 \\ & +2249 + 6634 + 5231 + 7232 \\ & +9921 + 9284 + 5647 + 4342 \\ & +8265 + 1124 = 92{,}271.\end{aligned}$$

Computing the paging cost is much easier. This can be simply achieved by counting the number of incoming calls in the LA, and, multiplying it by the number of cells (of the LA). In this case, the total number of incoming calls for the LA in Fig. 4a is given in Fig. 4b and would be

$$\begin{aligned}&\text{The number of paging transactions} \\ &\quad = (129 + 543 + 531 + 342 + 992 + 162 + 552) \times 7 \\ &\quad = 22{,}757.\end{aligned}$$

Finally, if this procedure is repeated for all LAs of the network, the total number of location updates and paging transactions can be calculated. For the marked LA in Fig. 4 these costs would be as follows:

$$\begin{aligned}&\text{Total number of location updates} = 92{,}271, \\ &\text{Total number of paging signals} = 22{,}757, \\ &\text{Total cost} = 92271 \times 10 + 22757 = 9{,}45{,}467.\end{aligned}$$

Note that in the above calculation, the cost of a location updates is assumed to be 10 times more expensive than that of a paging transaction (i.e. $\beta = 10$).

4. Generating a network and user profiles

The generation of network scenarios and profiles of mobile users are crucial steps for testing the proposed algorithms. A simulator developed by the authors of this work is used to test the algorithms. A brief description of the simulator is provided here for continuity, however, more details are provided elsewhere in the literature [45].

4.1. Network generation

This part of the simulator deals with the physical characteristics of a network, i.e. cell topology and cell attributes.

(1) *Cell topology*: Cells are divided into three classes according to their size, i.e. small, ordinary, and, large cells. The smaller cells represent 'crowded' cells that exist in any real-world network, for example, the downtown area cells that must

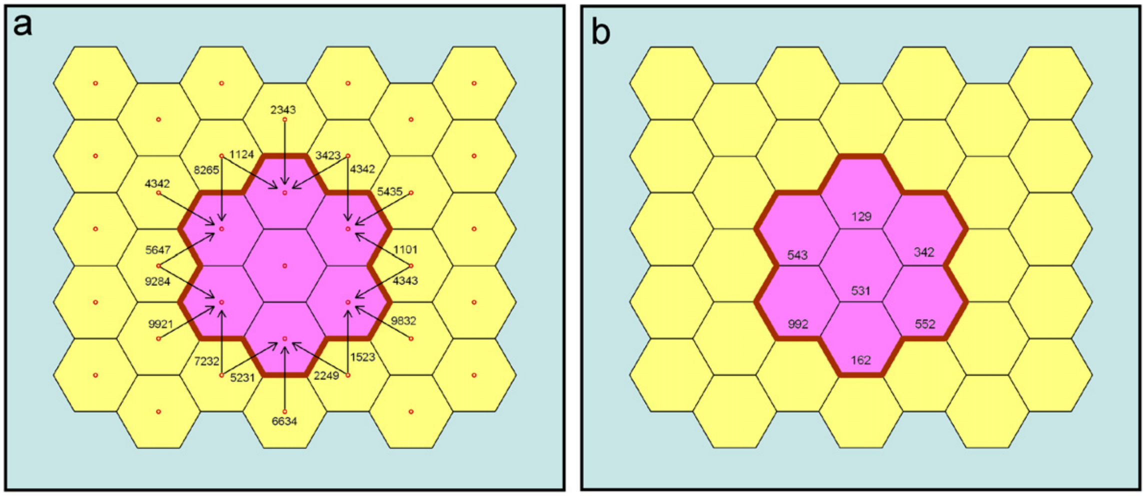

Fig. 4. (a) Entering flow and (b) call arrival for an instant LA.

process enormous numbers of users at business hours. Ordinary cells, on the other hand, represent cells that deal with normal or average traffic. In addition, large cells signify those areas of the network that may not have many users regularly, such as, parks or segments of highways.

(2) *Cell attributes*: Cells can also be categorized according to the nature of the area they cover. For example, in a real network, there are cells that cover areas in which people usually work or live in. Cells are categorized as, working cells, ordinary cells, and living cells. The 'working cells' class handles cells of the network that are busy during working hours. These cells are usually busy (i.e. crowded with users) in daytime and sparse at night time. 'Ordinary cells' are normal cells in terms of the number of people and the way they use the network, almost regardless of the time of the day. The 'living cells' class represent cells that do not have many users during the day while they tend to be busy during the evening.

4.2. Population generation

After classifying cells in terms of size and attributes, different user profiles need to be generated to match the above mentioned cells in term of their usage. Motivated by the experiences presented in [45], the behavior of users can be categorized using two distinct characteristics: movement and calling patterns.

(1) *Movement patterns*: In this work, users' movements are classified into day and night patterns. The movement patterns during daytime are classified into three subclasses: workers, ordinary, and commuters, and each of these classes has its own characteristics. For example, workers are a class of people who leave their homes at a certain time in the morning, with a predefined probability pattern, and return to their homes at a certain time. The ordinary class is assigned to users who have a regular pattern in their weekly behavior with some unpredictable movements. A typical example of such a class is a user who holds casual or part time jobs. Finally, the commuter class represents those users who always move around the network without a particular pattern, such as, taxi drivers. To simplify things, however, during night time it is assumed that almost all users have the same movement pattern.

(2) *Calling pattern*: This is another aspect that may vary from one user to another. Again, calling patterns can be classified into two classes for day and night times. Based on existing literature [39–41,21,8], it is believed that users follow a Bayesian probability as their calling pattern with different mean values.

4.3. User profile coding

The coding of a user profile is the last operation that has to be performed to render the simulator usable. The output profile of the network simulator is a number of complex timetables for every single user in the network. Each TimeTable contains several codes to demonstrate the user movement and calling behavior. As a result, working with such a huge database for each user could be really time consuming and full of redundant computations. Therefore, before applying the final algorithms for any section of a given TimeTable, several offline computations are performed to make the final output more usable. These computations deal with summarizing the TimeTable by transforming it into another time table. The output of such process, the EventTable, is helpful in minimizing the computation time for each user and eliminating the redundant processes by sorting the events that can be performed before launching the final location management algorithm. In fact, these computations can be completed during the idle modes when the system has spare time to perform the computations for each user.

Tables 1–3 show an instance of such computation before and after this coding. Tables 1 and 2 contain raw information about movement and calling behavior of a sample user, while Table 3 shows the more structured information. The cell crossing time (by a mobile terminal) is used to construct Table 3. For example, in Table 1, a particular user, was in positions (58,292) and (96,335) at 12.63 [12:38′] and 15.5 [15:30′], respectively,

Table 1
A sample TimeTable (movements) for an ordinary mobile user

Time (h)	Position
8.78	(96,335)
11.66	(58,292)
12.63	(58,292)
15.5	(96,335)
23.3	(96,335)
23.37	(105,331)
23.8	(105,331)
23.85	(96,335)

Table 2
A sample TimeTable (phone calls) for the mobile user in Table 1

Time (h)	Call duration	Call type
0.84	0.35	F
1.33	0.02	F
3.35	0.01	T
6.28	0.5	F
6.65	0.01	F
8.5	0.01	F
14.34	0.97	F
14.66	0.01	F
14.71	0.46	T
16.57	1.1	T
23.02	0.14	T

F: received call; T: made call.

Table 3
A EventTable for the mobile user in Tables 1 and 2

Time (h)	Event	Cell number
0	1	29
0.84	3	29
1.33	3	29
3.35	2	29
6.28	3	29
6.65	3	29
8.5	3	29
9.6	1	20
14.34	3	20
14.66	3	20
14.7	1	29
14.71	2	29
16.57	2	29
23.02	2	29
23.35	1	23
23.85	1	29

1: location update; 2: received call; 3: made call.

and made a phone call at 14.66 [14:40′] (Table 2). Note that, if the user profile is constructed for several days, then an individual EventTable is made for used for each day. Also, it is important to note that Table 3 is constructed after performing offline computations on Tables 1 and 2 to compress the size of data. For example, the information about the movement of the user from position (58,292) at 12.63 [12:38′] to (96,335) at 15.5 [15:30′] cannot be seen in Table 3 although a call the user placed at 14.66 [14:40′] can still be seen. It is worth noting that the EventTable only shows the time when the user actually crosses a cell and enters another (this is exactly when a location update is expected to be performed). For example, a user (in Table 1) passes cell 20, at coordinates (58,292), to cell 29, with coordinates (96,335), at a time between 12.63 and 15.5, respectively. However, the time calculated for this action is 14.7 as shown in Table 3.

5. Clustering users' patterns

The goal of this procedure is to extract patterns from users' movements to employ for future location prediction. If a user follows a particular pattern while moving, even for certain hours of the day, this can be used by the network to estimate the user's location without the need for explicit updates by the user. Two different approaches are used in this work: statistical and k-means clustering. The output of both clustering algorithms is a PatternTable which is similar to the EventTable but with an extra column known as 'Accuracy'. The accuracy factor, which is a figure between [0.0 1.0], represents the accuracy of the clustering at a particular time of the day. Fig. 5a and b show the movement patterns for two users. Fig. 6 shows that the accuracy factor for user of Fig. 5a is higher than that in Fig. 5b. This indicates that the location of the user in Fig. 5a is more predictable than that of Fig. 5b.

5.1. Statistical clustering

By using a statistical technique, based on the z-score, one can form clusters (of EventTables) by eliminating data that are apart. The mean value and the standard deviation of the sample data are calculated and used to purge data that are far from the mean (which is the cluster center). The following procedure demonstrates this clustering technique more clearly:

Step 1: Let $X = (x_1 x_2 \dots x_N)$ contains the input data to be clustered where N is the data size.

Step 2: Let $\mu_X = Mean(X) = \frac{\sum X}{N}$ and $\sigma_X^2 = Variance(X) = \frac{\sum (X-\mu_x)^2}{N}$.

Step 3: If $\mu_X/\sigma_X < 10$, then, $\sigma_X = \mu_X/10$.

Step 4: For all x_i's, if $|x_i - \mu_X|/N < 2.0$ then remove x_i from X.

Step 5: Repeat steps 2–4 until no x_i can be eliminated from X.

Step 6: Let $ClusterCenter = \mu_X$ and $\alpha_X = Accuracy(X) = \left(\frac{SizeOf(X)}{N}\right)^2$.

Note that, because the user profiles in this case contain two elements, (x, y) as the location of the user in the network, the above procedure is accomplished for any of these two elements separately and the cluster center is considered as (μ_x, μ_y). However, the cluster accuracy in this case is calculated as follows:

$$\alpha = \sqrt{\frac{\alpha_X^2 + \alpha_Y^2}{2}}.$$

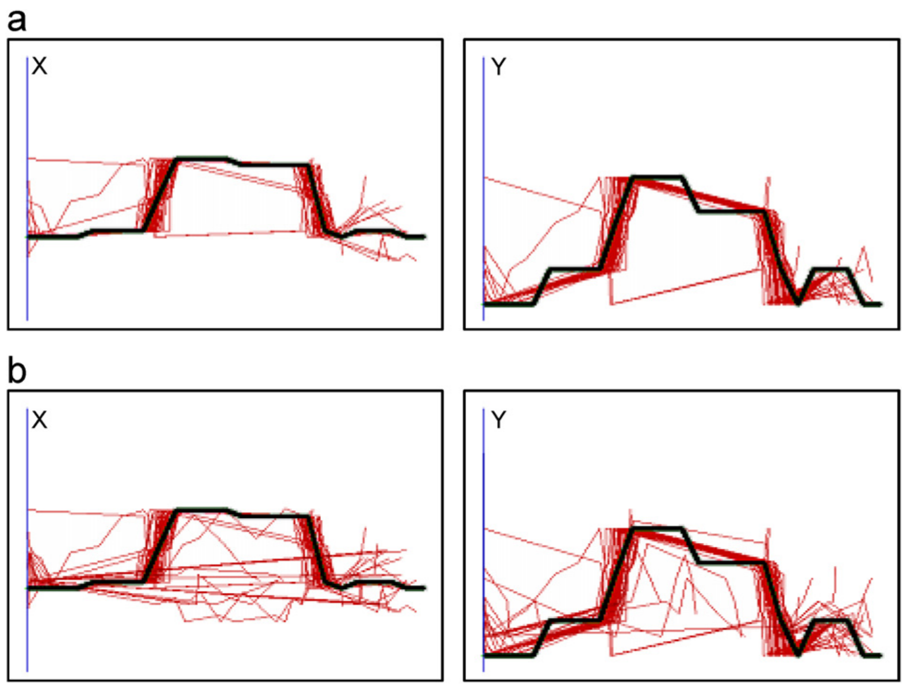

Fig. 5. Two similar movement profiles for two mobile terminals.

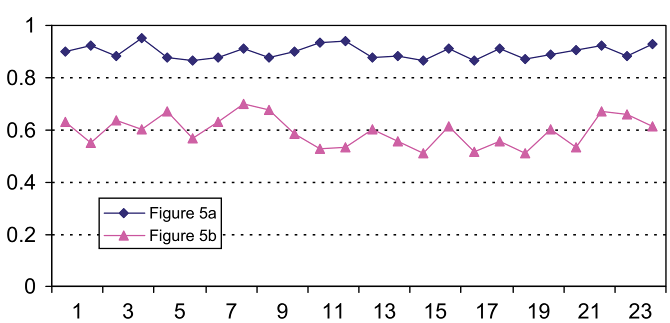

Fig. 6. Accuracy factor for the mobile terminals in Fig. 5.

5.2. K-means clustering

The k-means algorithm partitions a collection of n vectors $x_j : j = 1, \ldots, n$ into c groups $G_i : i = 1, \ldots, c$, and find a cluster center in each group such that a cost function of dissimilarity measure is minimized [24,49]. To achieve this, let:

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, x_k \in G_i} \|x_k - c_i\|^2 \right),$$

where $J_i = \sum_{k, x_k \in G_i} \|x_k - c_i\|^2$ is the cost function within group i.

The partitioned groups are typically defined by an $c \times n$ binary membership matrix U, where the element u_{ij} is '1' if the jth data which point x_j belongs to group i and '0' otherwise.

$$u_{ij} = \begin{cases} 1, & \|x_j - c_i\|^2 \leqslant \|x_j - c_k\|^2, \\ & \quad k \neq i, \\ 0 & \text{otherwise} \end{cases}$$

and the membership matrix U has the following properties:

1. $\sum_{i=1}^{c} u_{ij} = 1 \quad \forall j = 1, \ldots, n,$
2. $\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} = n.$

Finally, after each step we should update c_i's as

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \quad \text{where } |G_i| = \sum_{j=1}^{n} u_{ij}.$$

The performance of the k-means algorithm depends on the initial position of the cluster centers. However, as it can be seen, the x and y locations of a user's profile are considered simultaneously, where x_j is a two-dimensional vector containing the x and y location of the user at any time. The accuracy factor in this case is defined as the portion of the number of data points that make the largest cluster with respect to the number of all data points as follows:

$$\alpha = \left(\frac{N_{lu}}{N} \right)^{\lambda},$$

where the N_{lu} is the number of elements in the most crowded (dense) output cluster, N is the total number of data; and, λ is a factor to tune the accuracy factor of such clustering technique to get the best performance, and is set to 0.25 based on trial and error.

6. Location management strategies

As explained earlier, in static strategies, all information about the mobile network are gathered, and then passed to an optimization algorithm to find the optimal (or the most efficient) network configuration. Then, this network configuration is used for all users regardless of any changes in network operating conditions. Now, in the dynamic mode, the network configuration is constantly tuned to derive the best possible performance.

6.1. Static (offline) strategies

Several methods have been proposed to solve this problem based on neural networks, genetic algorithms, simulated annealing, and others [42–44]. However, these techniques have drawbacks, apart from the fact that they cannot be modified when the operating conditions change (e.g. increase in the number of users). Their performance is highly dependent on the quality of the data that was used to train the learning algorithm. However, these techniques can still provide valuable insight when designing mobile networks.

6.2. Dynamic (online) strategies

Dynamic strategies deal with network configurations that may change many times after network deployment. A network configuration is set in advance and during network operation the online algorithm constantly modifies the configuration based on users' behavior and deployment patterns. As a result, the network might change several times during the day and many more during the life of the network.

As it is the case with static techniques, metaheuristics can provide some of the best solutions for dynamic mobility management [16–23,25–27,35–37,46,47,30–33,9,8,5,3,12–14,3,50, 53,55,56]. Before describing the proposed approach in this paper several definitions need to be outlined.

(1) *Paging classes*: There are two different paging classes: non-hierarchical and hierarchical. To clarify this, assume that the network managed to pinpoint a user to be in an area that includes more than one cell. Now, the probability of the location of the user in any given cell is calculated for all the cells in this area and based on this information the network will attempt to locate the exact location of the user. In the hierarchical mode, the network pages the user in the most probable cell in the network first, and then, if unsuccessful, it tries other cells, which usually surround the first cell based on some probability. This procedure continues until reaching a certain number of iterations or the user is located. In non–hierarchical paging, the user is paged in all the possible cells simultaneously.

Fig. 7 shows an example of hierarchical and non–hierarchical paging. In this example, the network tries to locate a user who is expected to be in cell S1. The user is paged in this cell first, and if unsuccessful, the network pages all cells marked as S2 in the second phase of paging. If this also fails, then the user would be paged in all cells marked as S3. However, if the network employs non-hierarchical paging, then the network will page the user in all cells marked as S1, S2, and S3, simultaneously. In this case the network paging ends at level S3 for both classes of paging.

(2) *Cell based user pattern extraction*: To extract a movement pattern, the network collects information about the user on different days and then attempts to find a pattern in the data. However, because mobile users are not equipped with GPS or any other positioning system, which can report the exact location, the network geographic cell information is used to construct user's profiles. Therefore, every time a user updates its location when moving between cells, the new cell is logged. For simulation purposes the location of the user is approximated by the coordinates at the center of the cell. The same applies when entering a LA. The user location is approximated to be the center of the LA.

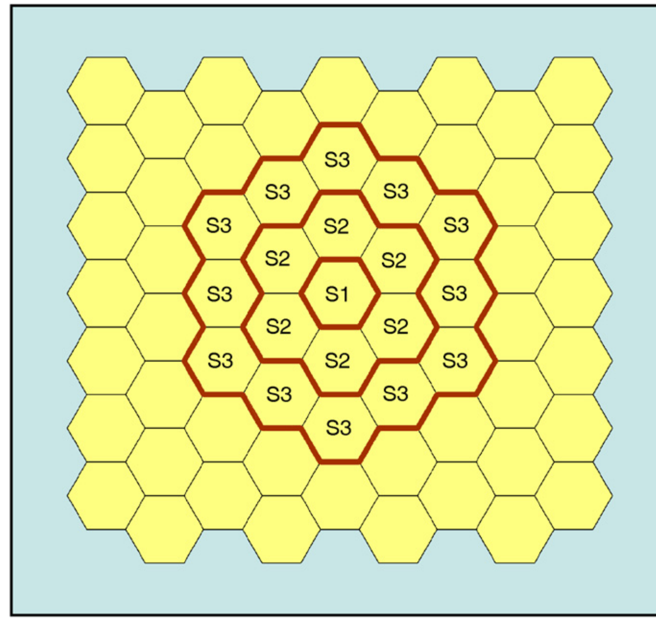

Fig. 7. Hierarchical/non-hierarchical paging cells for a sample network.

(3) *Cell or LA based paging*: Similarly, when the network attempts to locate a user, it can page on a Cell-by-Cell or a LA-by-LA basis. In the Cell-by-Cell case, the network tries to locate the user's current cell, while in the LA-by-LA case, the network attempts to locate the user's LA.

(4) *Paging criterion*: When a user's pattern is computed based on a given profile, an accuracy value is assigned to each section of the profile. To explain this, refer to the two profiles depicted in Fig. 5. It is assumed that the (x, y) coordinates of the user are gathered for a period of 10 days. In this case, the solid line could be the pattern that the user mostly follows. As it can be seen, in Fig. 5a, the solid line matches almost all the cases, while in case (b) the solid line partially matches the user's movements. The accuracy factor in this case distinguishes these two profiles from each other. Therefore, the accuracy value is assigned for each section of this time line to represent how accurate they are. For example, the lines depicted in Figs. 6a and 6b can represent the accuracy of the pattern extracted in Figs. 5a and 5b, respectively.

In this paper, three different criteria are used to ensure mobile terminals to update their locations: always update (just like GSM), update based on the current accuracy, and update based on a fixed 0.25 value (i.e. value of the accuracy of pattern following). In all these cases, the mobile terminal is forced to update its location when the rate of the pattern following is less than the accuracy of the paging criterion used at the time.

7. Dynamic location management strategy

A general overview of the approach is given in Fig. 8. To create a robust user profile data needs to be collected over several days to extract a movement pattern. Then, a pattern is used to approximate the user's movements in the days that follow.

7.1. Pattern extraction

Two different approaches are used to extract movement patterns: statistical and k-means clustering. In the statistical approach, which is designed based on the z-score technique, the x and y locations of the user are gathered for several days, then, a statistical technique is used to estimate the x and y locations of the user for different time intervals. For example, if the time interval is set to 1 h, it means that the statistical classifier is used to determine the location of the user every 1 h. On the other hand, because the clustering technique in this case is supposed to be one-dimensional, the x and y variable of each user's profile is supposed to be independent. On the other hand, k-means clustering classifies the user's information in a completely different way. In this case, the x and y locations of the user are considered simultaneously during clustering.

7.2. Online location management strategy

The online location management strategy is designed to trace the movements of the users, extract a pattern out of their movement history, upload this information into each mobile terminal, and force mobile terminals to update their location if they are not following the predicted pattern. The following procedure shows how the overall algorithm works. This procedure is launched for every single user in the network. This procedure has two components: updating and paging. The updating phase has two components, which is performed by the network and the mobile terminal.

Update phase (*at the network level*):
Step 1: Collect profile data for a given user.
Step 2: Extract a pattern from the profile for different time slots with an 'accuracy factor' for each point.
Step 3: Construct a PatternTable from the above information.
Step 4: Upload this PatternTable into the mobile terminal.

Update phase (*at the mobile terminal level*)
Step 1: Store the PatternTable uploaded from the network.
Step 2: Calculate the PatternFollowingRate whenever the terminal crosses a cell (this factor shows how close the mobile terminal is following the pattern calculated by the network).
Step 3: Calculate the accuracy of the pattern at the time of crossing a cell, i.e. ExpectedAccuracy.

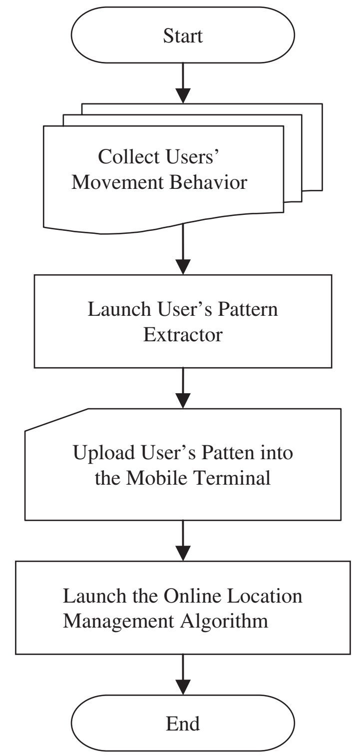

Fig. 8. The dynamic location management strategy.

Step 4: if ExpectedAccuracy < PatternFollowingRate then the mobile terminal should update its position; otherwise, cross the cell without performing a location update.

Paging phase (*hierarchical mode*) (*at the network level*):
Step 1: Find the cell (S1) that the network expects the user to be in (this could be in the PatternTable or the last updated location).
Step 2: Page the user in S1.
Step 3: If the user cannot be found in S1 then compute all locations in S2 (in all cells surrounding S1) as given in Fig. 7.
Step 4: Page the user in S2.
Step 5: If the user cannot be found in S2 then compute all locations in S3 (in all cells surrounding S2) as given in Fig. 7.
Step 6: Page the user in S3.
Step 7: If the user cannot be found in S3 then inform the network that the user cannot be located.

Note that, in the above procedures few considerations must be taken into account. The manner in which the users' profiles are constructed from their movements is based on what the network supports. For example, if they are GSM mobile users, then the only information that can be used or collected is the location of the LA. However, the users' profile can be calculated on a Cell-by-Cell basis if it is supported by the network.

Also, the PatternFollowingRate is a value between [0, 1] and is calculated based on the cell/LA number. For example, if the current cell/LA is the same as the one estimated by the PatternTable, then this figure is set to 1.0. Otherwise, the farther the current cell/LA is from the cell/LA, which is estimated by the PatternTable, the closer this figure is to 0.0. To achieve this, time is used for calculating this factor, as shown below.

Step 1: Let RealPosition and ExpectedPosition be the real location of the user calculated from the PatternTable for a specific time.

Step 2: Let dist=Distance(RealPosition,ExpectedPosition).

Step 3: Find the time that the user will be at the user's RealPosition based on the PatternTable, which is called MatchingHour.

Step 4: If the user never passes the RealPosition based on the PatternTable data, then PatternFollowingRate is set to 0.0.

Step 5: If the user passes the position RealPosition based on the PatternTable data at time T, then, PatternFollowingRate= 1 − AbsoluteValue(MatchingHour-T) *UpdatingProbabilityIncreaseFactorPerHour .

In the above algorithm, UpdatingProbabilityIncreaseFactorPerHour is a factor that is used to judge whether the user should perform a location update or not. A higher value of UpdatingProbabilityIncreaseFactorPerHour forces the mobile terminal to update its location even if it is not following the estimated pattern for a short period of time (e.g. several minutes). Note that, in some cases when a mobile terminal is not forced to update its location (to save bandwidth); a situation arises in which the network loses track of the terminal. However, this is quite rare due to the efficient use of the UpdatingProbabilityIncreaseFactorPerHour parameter.

The ExpectedAccuracy factor is selected based on the way that the network forces the mobile terminal to update its location when not following the PatternTable. There are three options: always update, accuracy, and fixed (0.25). In the always update case, whenever the mobile terminal crosses a cell/LA it updates its location. In this case, the mobile terminal is assumed to follow the GSM style of location update. In the accuracy case, the mobile terminal is driven to update is location information if the accuracy is less than what the network has estimated (based on their PatternTable). Note that, the network uploads the PatternTable into each and every mobile terminal, which allows each terminal to perform the same computation that the network can perform at all times. Thus, when crossing a cell/LA each mobile terminal is able to determine the expected position that the network calculates and if the accuracy is less than the one predicted by the network then the terminal is forced to update its location. Finally, in the fixed (0.25) case, whenever the accuracy of finding the user is less than a given value, such as 0.25 in this case (determined by trial and error), the mobile terminal is forced to update its location.

The paging mechanism in this work employs three hierarchical levels, however, the algorithm can be easily modified to include more or less levels depending on the design specifics.

8. Simulation results

In this section, two test networks are simulated to demonstrate the performance of the algorithm proposed in this work. The examples are generated to benchmark against GSM type of networks. The users' profiles for each test network are generated by a procedure developed elsewhere in the literature [45]. In each test network, the location management is handled by different approaches, clustering techniques (statistical and k-means), paging techniques (non-hierarchical and hierarchical), paging criteria (always update, accuracy and fixed (0.25)) and finally different cell bases (cells and LAs). In all the cases, the results are compared with those that could have been generated by a GSM network. Each of the test networks is examined using two scenarios: LA scenario and cell–LA scenario. In the LA scenario, information related to mobile users are collected based on the LA-by-LA scheme, and then the users are paged on LA-by-LA basis as well (GSM like). On the other hand, in the cell–LA scenario, user data is collected based on the LA-by-LA scheme, and then users are paged on a cell-by-cell basis. This ability to switch between different schemes makes the proposed approach more adaptable.

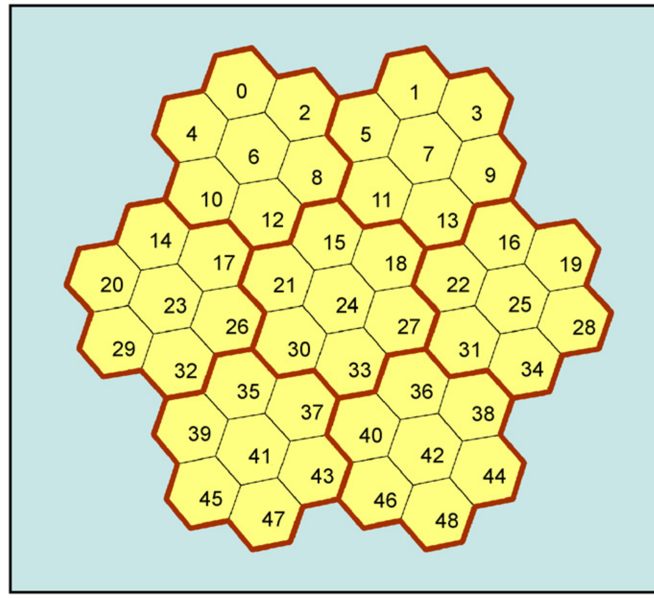

Fig. 9. A test network of 19 LAs.

Table 4
Location update scheme for the 19-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Total number of location updates	130,751	39,934	54,465	40,494	57,278
Saved location updates	0	90,817	76,286	90,257	73,473

8.1. Test-network (19-LAs)

The network consists of 19 LAs (19×7=133 cells) as shown in Fig. 9. The LAs are arranged like a GSM network. A total number of 1266 user profiles are generated who use the network for 50 days according to the procedure proposed in [45].

The overall experiment is conducted in the following fashion. The network is deployed as a normal GSM network for the first

Table 5
Successful paging using the hierarchical mode for the 19-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Successful paging-1	316,792	296,997	304,605	297,221	306,568
Successful paging-2	0	17,185	10,620	16,628	9022
Successful paging-3	0	2303	1378	2427	993
Unreachable customers	0	307	189	516	209

Table 6
Cost of the network for the 19-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Cost H-1	3,525,054	2,616,884	2,762,194	2,622,484	2,790,324
Cost H-2	3,525,054	3,329,617	3,208,045	3,313,342	3,158,706
Cost H-3	3,525,054	3,451,158	3,283,624	3,443,409	3,214,566
Cost NH-1	3,525,054	2,616,884	2,762,194	2,622,484	2,790,324
Cost NH-2	3,525,054	12,647,352	12,793,992	12,635,067	12,803,544

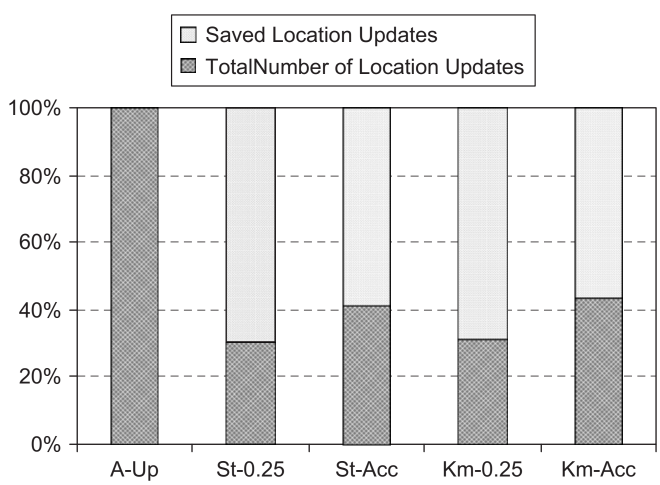

Fig. 10. Saved location Updates for the 19-LA network using the LA scenario.

10 days, and then, the profiles are collected for this period and used to extract the behavior patterns. These behavior patterns are used to check the performance of the algorithm for the remaining 40 days (days 11–50). Tables 4–6 show the results for different LA scenarios. In each table, the results are compared with a GSM network (listed as ‘always update’ in the tables) with a percentile value representing the performance of specific parameters in the algorithm. Also, Figs. 10–15 demonstrates the performance of the algorithm as compared with a GSM network. Tables 7–9 (Figs. 16–21) show the performance for the cell–LA scenario.

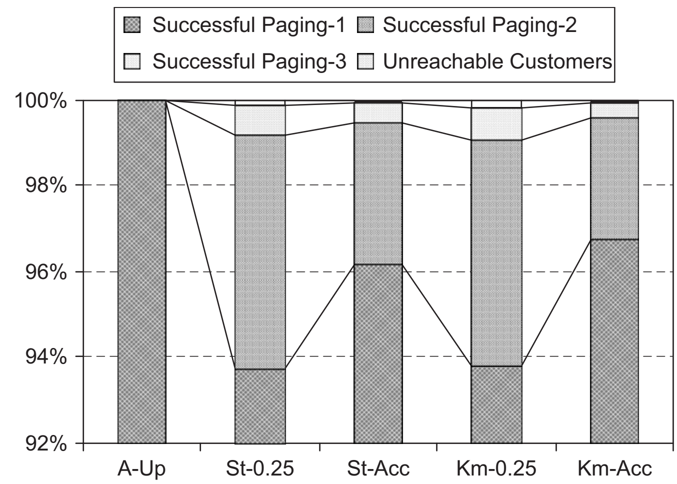

Fig. 11. Successful paging in the hierarchical mode for the19-LA network using the LA scenario.

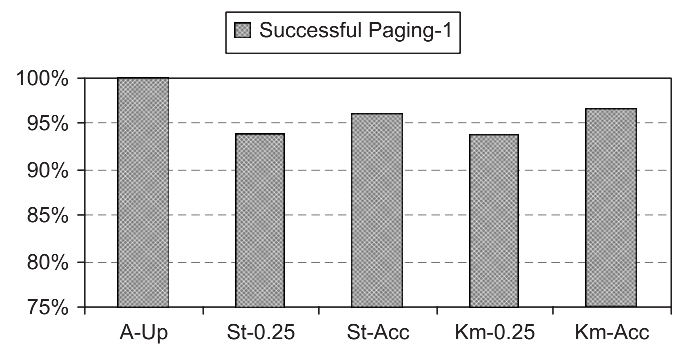

Fig. 12. One level successful paging for the 19-LA network using the LA scenario.

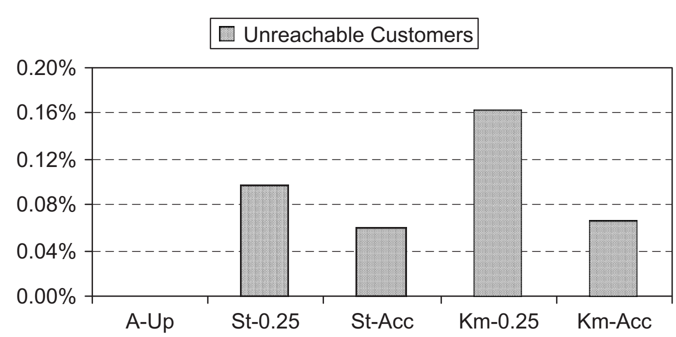

Fig. 13. Unreachable users after three hierarchical paging levels for the 19-LA network using the LA scenario.

8.2. Test-network (51-LAs)

This network consists of 51 LAs ($51 \times 7 = 357$ cells) as shown in Fig. 22. The LAs are arranged like a GSM network.

A total number of 1785 user profiles are generated to use the network for a period of 50 days similar to the conditions used for the previous experiment. The results are given in Tables 10–15 and Figs. 23–34.

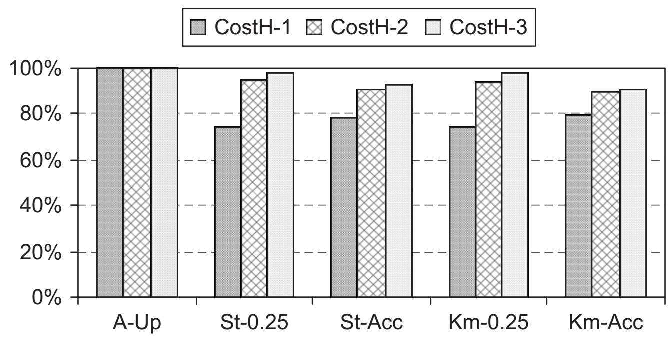

Fig. 14. Cost of the network in the hierarchical mode for the 19-LA network using the LA scenario.

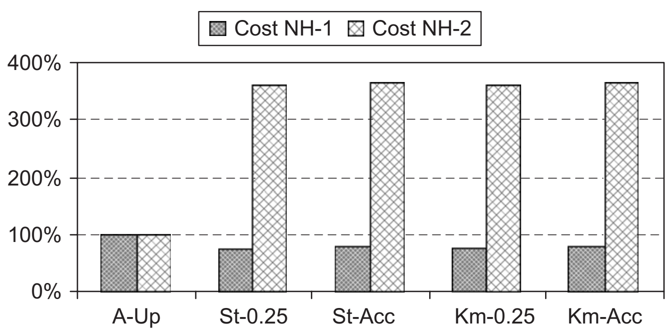

Fig. 15. Cost of the network in the non-hierarchical mode for the 19-LA network using the LA scenario.

Table 7
Location update scheme for the 19-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Total number of location updates	320,941	166,769	237,644	171,781	252,432
Saved location updates	0	154,172	83,297	149,160	68,509

Table 8
Successful paging using the hierarchical mode for the 19-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Successful paging-1	316,792	274,133	303,067	274,550	306,821
Successful paging-2	0	31,344	8758	31,019	7151
Successful paging-3	0	5605	1920	5146	895
Unreachable customers	0	5710	3047	6077	1925

Table 9
Cost of the network for the 19-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Cost H-1	3,526,202	1,984,482	2,693,232	2,034,602	2,841,112
Cost H-2	3,526,202	2,239,663	2,775,251	2,287,704	2,900,848
Cost H-3	3,526,202	2,361,254	2,830,772	2,401,665	2,931,064
Cost NH-1	3,526,202	1,984,482	2,693,232	2,034,602	2,841,112
Cost NH-2	3,526,202	3,708,074	4,406,865	3,758,583	4,553,596
Cost NH-3	3,526,202	6,768,759	7,483,384	6,807,546	7,625,225

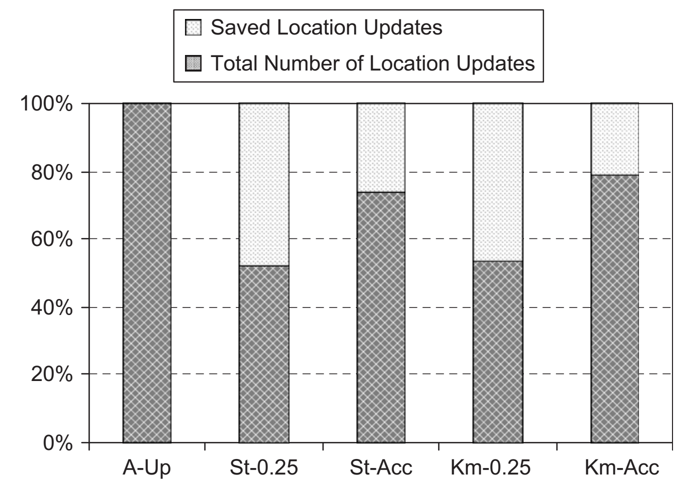

Fig. 16. Saved location updates for the 19-LA network using the cell–LA scenario.

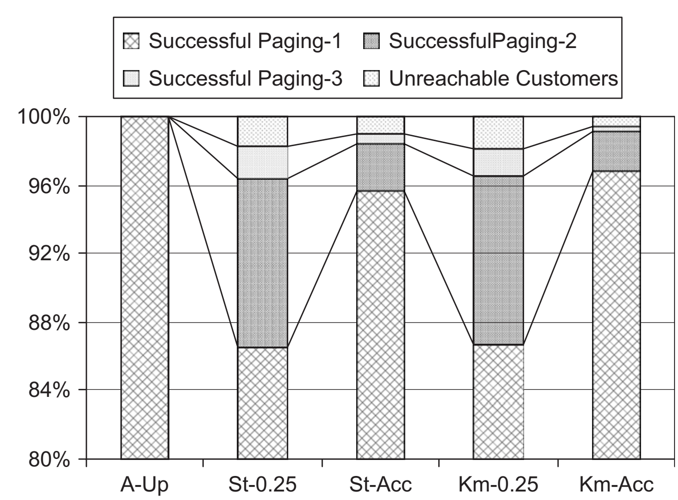

Fig. 17. Successful paging in the hierarchical mode for the 19-LA network using the cell–LA scenario.

9. Discussion

Several observations can be made about the results that were generated by the algorithms proposed in this work. These are listed below.

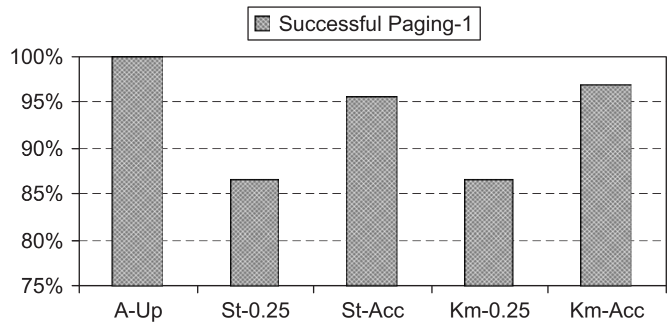

Fig. 18. One level successful paging for the 19-LA network using the cell–LA scenario.

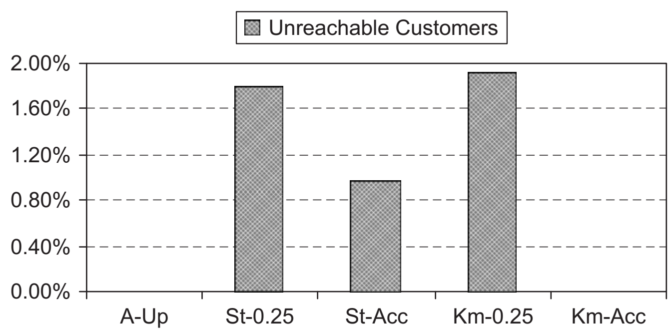

Fig. 19. Unreachable users after three hierarchical paging levels for the 19-LA network using the cell–LA scenario.

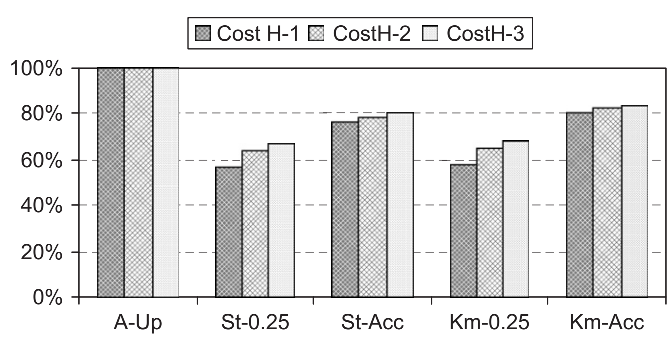

Fig. 20. Cost of the network in the hierarchical mode for the 19-LA network using the cell–LA scenario.

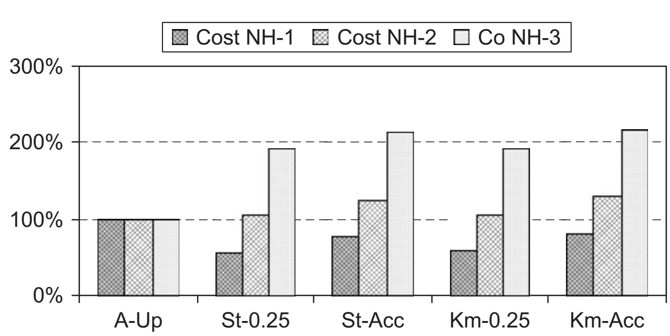

Fig. 21. Cost of the network in the non-hierarchical mode for the 19-LA network using the cell–LA scenario.

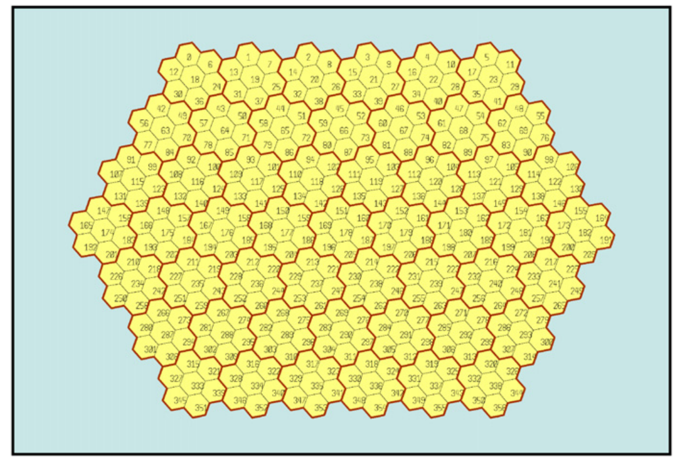

Fig. 22. A test network of 51 LAs.

Table 10
Location update scheme for the 51-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Total number of location updates	220,938	80,539	103,662	81,810	108,006
Saved location updates	0	140,399	117,276	139,128	112,932

Table 11
Successful paging using the hierarchical mode for the 51-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Successful paging-1	447,348	429,137	436,388	429,714	438,052
Successful paging-2	0	14,452	8333	13,593	7638
Successful paging-3	0	2656	1609	2521	944
Unreachable customers	0	1103	1018	1520	714

9.1. Cell based scenarios

In the cell–LA scenarios, the results are compared with the ‘always update’ case for each test network, which is equivalent to the Cell-by-Cell always update case. However, by comparing Tables 6–9 and Tables 12–15 for the 19-LA and 51-LA, respectively, it can be seen that the always update costs for the Cell-by-Cell and the LA-by-LA (GSM like) scenarios are roughly equal. The reason of this is the huge number of calling transactions, which is roughly between 5 and 20 calls per day, for each user in our simulation. In fact, this figure is much less for real networks and that is why mobile networks predominantly employ GSM based topologies. Thus, and without loss of generality, conclusions based upon comparing result for the always update scheme in the Cell-by-Cell basis can be

Table 12
Cost of the network for the 51-LA network (LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Cost H-1	5,340,816	3,936,826	4,168,056	3,949,536	4,211,496
Cost H-2	5,340,816	4,648,089	4,600,285	4,627,472	4,570,743
Cost H-3	5,340,816	4,905,696	4,791,965	4,882,363	4,680,188
Cost NH-1	5,340,816	3,936,826	4,168,056	3,949,536	4,211,496
Cost NH-2	5,340,816	19,925,120	20,160,123	19,921,079	20,184,999

Table 13
Location update scheme for the 51-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Total number of location updates	519,334	284,807	369,962	294,222	391,077
Saved location updates	0	234,527	149,372	225,112	128,257

Table 14
Successful paging in the hierarchical mode for the 51-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Successful paging-1	447,348	415,462	437,035	415,015	438,720
Successful paging-2	0	20,736	5155	21,643	5233
Successful paging-3	0	4557	1246	4318	1008
Unreachable customers	0	6593	3912	6372	2387

Table 15
Cost of the network for the 51-LA network (cell–LA scenario)

	Always update	St-0.25	St-Acc	Km-0.25	Km-Acc
Cost H-1	5,640,688	3,295,418	4,146,968	3,389,568	4,358,118
Cost H-2	5,640,688	3,486,214	4,208,687	3,583,357	4,409,860
Cost H-3	5,640,688	3,613,737	4,269,454	3,702,124	4,448,452
Cost NH-1	5,640,688	3,295,418	4,146,968	3,389,568	4,358,118
Cost NH-2	5,640,688	5,814,537	6,661,032	5,908,963	6,871,635
Cost NH-3	5,640,688	10,549,649	11,401,313	10,636,272	11,606,502

generalized to those of the LA-by-LA (GSM like). Therefore, in LA scenarios, the network completely follows the GSM protocols for location management, while for cell–LA scenarios, the network collects the users' information on LA-by-LA basis and then this information is used to page users on Cell-by-Cell basis.

9.2. Network cost

The cost associated with different cell scenarios are summarized in Table 16. The reason for having a relatively equal cost

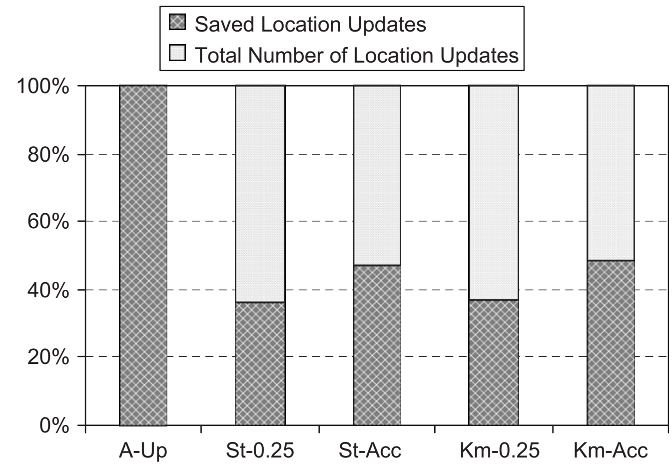

Fig. 23. Saved location updates for the 51-LA network using the LA scenario.

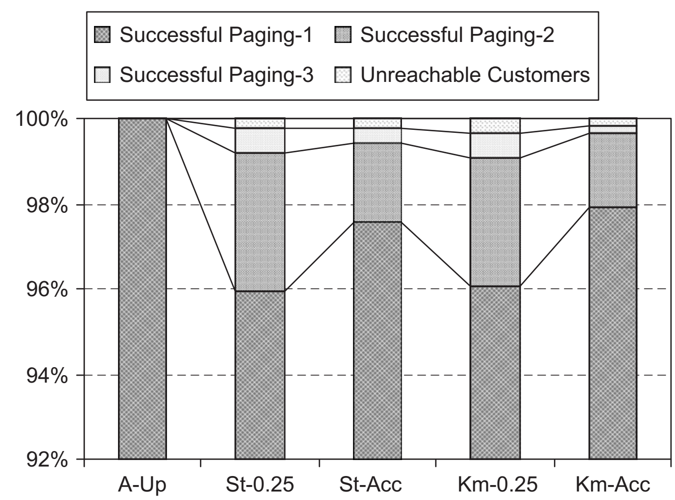

Fig. 24. Successful paging in the hierarchical mode for the 51-LA network using the LA-scenario.

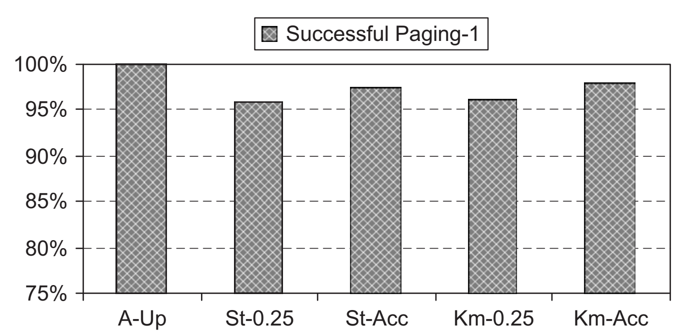

Fig. 25. One level successful paging for the 51-LA network using the LA scenario.

with the always update (roughly GSM like) for the 2-level non-hierarchical mode in the cell–LA scenario is because of the paging method used by the network. In this scenario a user is paged in seven cells simultaneously in the network whenever the network diverts an incoming call. Also, another reason for having such a relatively high cost for the 2-level non-hierarchical mode

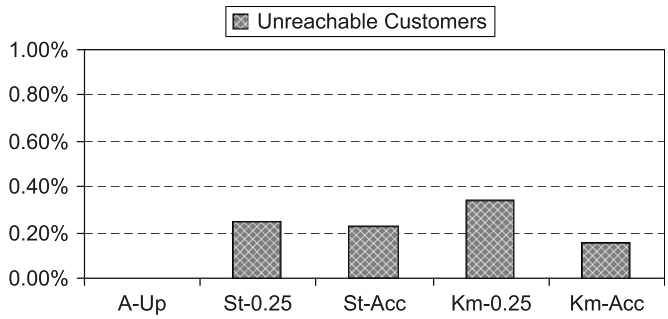

Fig. 26. Unreachable users after three hierarchical paging levels for the 51-LA network using the LA scenario.

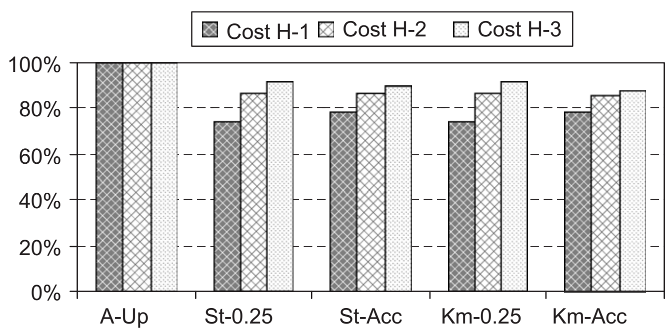

Fig. 27. Cost of the network in the hierarchical mode for the 51-LA network using the LA scenario.

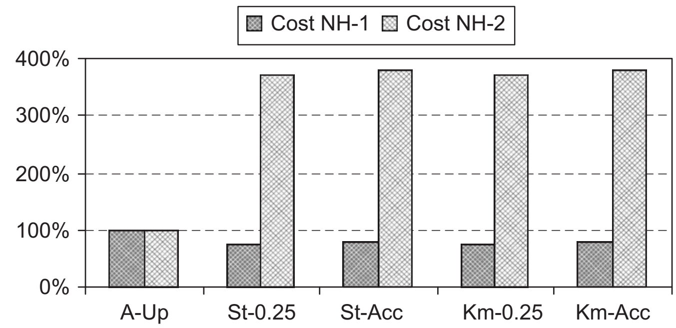

Fig. 28. Cost of the network in the non-hierarchical mode for the 51-LA network using the LA scenario.

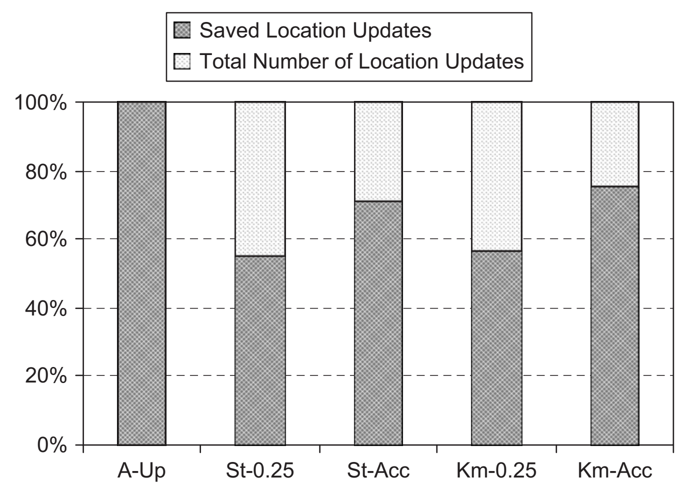

Fig. 29. Saved location updates for the 51-LA network using the cell–LA scenario.

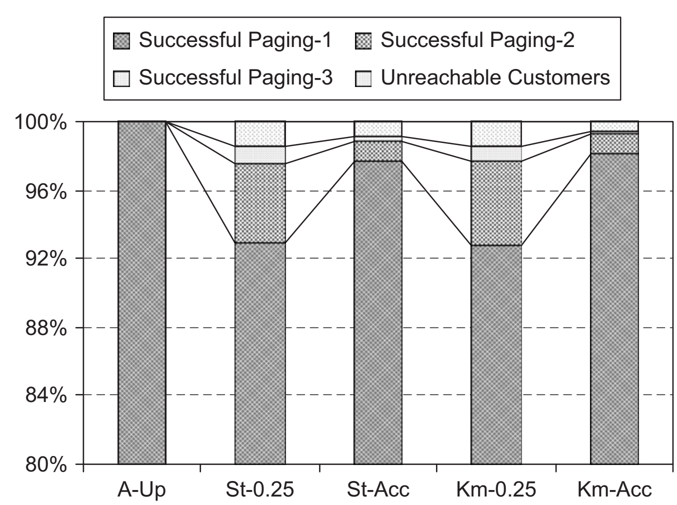

Fig. 30. Successful paging in the hierarchical mode for the 51-LA network using the cell–LA scenario.

in the LA scenario is the inefficiency of paging the users in the network. This means that there only one successful paging and $7 \times 7 - 1 = 48$ unsuccessful ones for each incoming call (note that, each LA has six neighboring LAs with a total number of 49 cells).

Similarly, the reason for having an expensive cost for the hierarchical mode with 3-level paging in the cell–LA scenario is because the user is paged in many irrelevant cells whenever the network attempts to divert a call. This means that for each successful paging there would be $1 + 6 + 12 - 1 = 18$ (the number of S1, S2 and S3 cells in Fig. 7) unsuccessful ones.

9.3. Saved location updates

The number of location updates that are performed in each of the above simulations is around 35% and 60% of the number of location updates that the network would go through if it follows the always update technique for the LA and cell–LA scenarios, respectively. This figure can be used to evaluate the performance of the pattern extraction techniques proposed in this work. It can be noted that the pattern extraction techniques managed to save nearly half of the updates that the network could have experienced.

9.4. Successful paging

It is also important to note that in almost 96% and 88% of the occasions, the network managed to locate the user in the first attempt in the LA and cell–LA scenarios, respectively. Further, 5–10% of the users that could not be located during the first attempt were located during the second one. In less than 1.5% of the occasions the network was unable to locate the user in the cell–LA scenario, while this value was around 0.3% for the LA scenario.

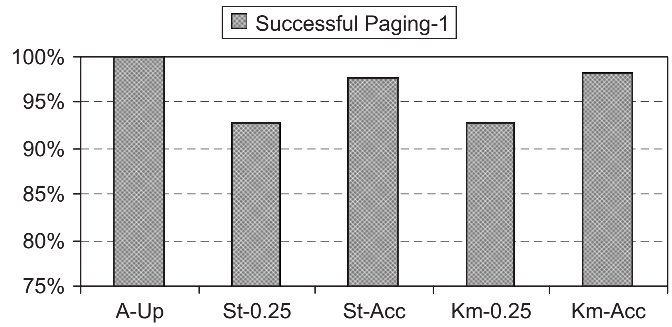

Fig. 31. One level successful paging for the 51-LA network using the cell–LA scenario.

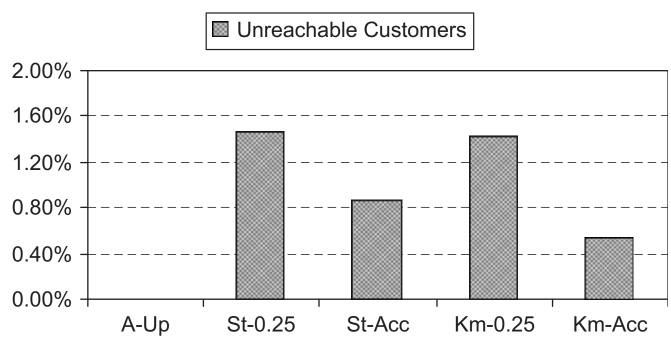

Fig. 32. Unreachable customers after three hierarchical paging levels for the 51-LA network using the cell–LA scenario.

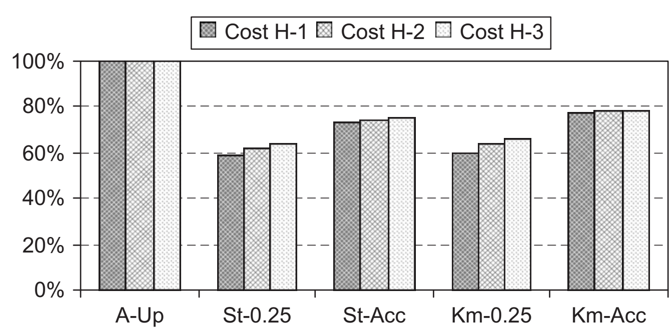

Fig. 33. Cost of the network in the hierarchical mode for the 51-LA network in the cell–LA scenario.

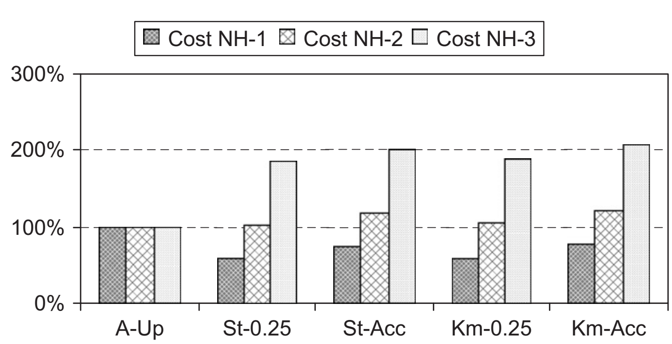

Fig. 34. Cost of the network in the non-hierarchical mode for the 51-LA network in the cell–LA scenario.

Table 16
Summarized network cost for different cell scenarios

	Cost H-1 (%)	Cost H-2 (%)	Cost H-3 (%)	Cost NH-2 (%)	Cost NH-3 (%)
LA-scenario	75	90	95	350	
Cell-LA-scenario	60	70	75	100	175

9.5. Accuracy versus Fixed (0.25) paging criteria

The accuracy paging criterion leads to a 10% saving in the number of location updates when compared to the Fixed (0.25) case. In addition, the network is more successful in locating users in during the first attempt when using the accuracy criterion, which leads to a slightly better performance than the Fixed (0.25) scenario.

9.6. Clustering techniques

It can be seen that there is not a major difference in performance between the different clustering techniques (statistical and *k*-means).

9.7. Comparing LA and cell–LA based scenarios

(1) *Network cost*: It has been noted that the cost of network management is almost 40% less for the cell–LA scenarios. The main reason for this considerable difference between the two scenarios is the fact that cell–LA scenarios experience less unsuccessful paging transactions when a network attempts to connect a call.

(2) *Saved location updates*: The LA scenarios usually lead to 20–30% less location updates. This difference is due to the following fact. In LA scenarios, the network forces users to update their location information whenever they change their current LA. In contrast, in cell–LA scenarios, users are expected to update their location information when they violate their Cell-by-Cell location strategy. This leads to less updates for LA scenarios.

(3) *Successful paging*: It has been shown that the success of locating a user during the first attempt is almost 6% better in LA scenarios when compared to similar setups in cell–LA scenarios. This is rather logical because the likelihood of finding a user in a paged LA (which contains seven cells) will be much higher than that of paging a single cell. However, it is still remarkable that the network managed to locate the exact cell of the user in almost 90% of times in the cell–LA scenarios.

(4) *Accuracy versus Fixed*: (*0.25*) *paging criteria*: The accuracy criterion leads to more saved location updates. The number of saved updates is around 30% higher when the network employs the LA scenario when compared to a similar setup of the cell–LA scenario. This figure is around 20% for the Fixed (0.25) case. In addition, the percentage of successful paging during the first attempt for the accuracy case seems closer to

the similar configurations in the LA scenario than that of cell–LA ones.

10. Conclusion

This paper presented a novel approach for solving the dynamic location management problem. Several clustering techniques were used to extract information from movement patterns of mobile users. This information was then used to predict users' behavior (motion) to reduce the number of location updates that need to be performed. Several distinct observations were made that articulated the results of extensive sets of simulation studies.

References

[1] D.P. Agrawal, Q-A. Zeng, Introduction to Wireless and Mobile Systems, Thomson Brooks/Cole Inc., 2003.
[2] I.F. Akyildiz, J.S.M. Ho, Movement based location update and selective paging for PCS networks, IEEE/ACM Transactions on Networking 4 (1996) 629–638.
[3] I.F. Akyildiz, W. Wang, A dynamic location management scheme for next-generation multitier PCS systems, IEEE Transactions on Wireless Communications, January 2002, vol. 1(1) pp. 178–189.
[4] K. Al-Tawil, A. Akrami, A new authentication protocol for roaming users in GSM networks, EEE International Symposium on Computers and Communications, 1999, pp. 93–99.
[5] K. Al-Tawil, A. Akrami, H. Youssef, A new authentication protocol for GSM networks, 23rd Annual Conference on Local Computer Networks, LCN '98, 11–14 October 1998, pp. 21–30.
[6] M.D. Austin, G.L. Stuber, Direction based handoff algorithms for urban microcells, Proceedings of the IEEE VTC-94, June 1994, pp. 101–105.
[7] N.A. Bar, I. Kessler, M. Sidi, Mobile users: To update or not to update?, Wireless Networks 1 (1995) 175–185.
[8] Y. Bejerano, I. Cidon, Efficient location management based on moving location areas, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, vol. 1, pp. 3–12.
[9] A. Bera, N. Das, Performance analysis of dynamic location updation strategies for mobile users, 20th International Conference on Distributed Computing Systems, 2000, pp. 428–435.
[10] G.V. Casares, O.J. Mataix, On movement based mobility tracking—an enhanced version, IEEE Communication Letters 2 (1998) 45–47.
[11] A. Catovic, S. Tekinay, Geolocation updating schemes for location aware services in wireless networks, Military Communications Conference, IEEE Communications for Network-Centric Operations: Creating the Information Force, MILCOM 2001, vol. 1, 2001, pp. 378–382.
[12] K.T. Chen, S.L. Su, R.F. Chang, Design and analysis of dynamic mobility tracking in wireless personal communication networks, IEEE Transactions on Vehicular Technology, vol. 51(3), May 2002, pp. 486–497.
[13] W.-J. Choi, S. Tekinay, An adaptive location registration scheme with dynamic mobility classification, IEEE International Conference on Communications, ICC 2002, vol. 1, 2002, pp. 440–444.
[14] J.X. Ci, W.-T. Tsai, A location transparent and cost effective routing algorithm for mobile component ware interaction and coordination, 21st IEEE International Performance, Computing, and Communications Conference, 2002, pp. 347–351.
[15] I. Demirkol, C. Ersoy, M.U. Caglayan, H. Delic, Location area planning in cellular networks using simulated annealing, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, vol. 1, 2001, pp. 13–20.
[16] F. Erbas, J. Steuer, D. Eggesieker, K. Kyamakya, K. Jobinann, A regular path recognition method and prediction of user movements in wireless networks, IEEE VTS 54th Vehicular Technology Conference, VTC 2001, vol. 4, Fall 2001, pp. 2672–2676.
[17] D. Gu, S.S. Rappaport, A dynamic location tracking strategy for mobile communication systems, 48th IEEE Vehicular Technology Conference, VTC 98, vol. 1, 18–21 May 1998, pp. 259–263.
[18] Il. Han, S.-S. Choi, D.-H. Cho, An automatic drawing method for location tracking in wireless networks, IEEE VTS 53rd Vehicular Technology Conference, VTC 2001, vol. 4, Spring 2001, pp. 2555–2559.
[19] J.S.M. Ho, J. Xu, History-based location tracking for personal communications networks, 48th IEEE Vehicular Technology Conference, VTC 98, vol. 1, 18–21 May 1998, pp. 244–248.
[20] N.S. Satish Jamadagni, Dealing with location uncertainty in mobile networks using contextual fuzzy cognitive maps as spatial decision support systems, IEEE 52nd VTS Vehicular Technology Conference, VTC 2000, vol. 3, 2000, pp. 1489–1492.
[21] D.G. Jeong, W.S. Jeon, Effective location management strategy based on user mobility classes, The Bridge to Global Integration Global Telecommunications Conference, GLOBECOM 98, vol. 3, IEEE 1998, pp. 1426–1430.
[22] X. Jiang, I.F. Akyildiz, A novel distributed dynamic location management scheme for minimizing signaling costs in Mobile IP, IEEE Transactions on Mobile Computing, vol. 1(3) July–September 2002, pp. 163–175.
[23] L. Jie, H. Kameda, L. Keqin, Optimal dynamic location update for PCS networks, Proceedings of 19th IEEE International Conference on Distributed Computing Systems, 1998.
[24] L.-J. Kau, Adaptive predictor with dynamic fuzzy K-means clustering for lossless image coding, The 12th IEEE International Conference on Fuzzy Systems, FUZZ '03, vol. 2, 25–28 May 2003, 944–949.
[25] V. Krishnamurthy, S. Singh, Sequential paging of mobile users in GSM cellular networks-a POMDP approach, IEEE Global Telecommunications Conference, GLOBECOM '00, vol. 1, 2000, pp. 182–187.
[26] B.-K. Lee, C.-S. Hwang, A predictive paging scheme based on the movement direction of a mobile host, IEEE VTS 50th Vehicular Technology Conference, VTC 1999, vol. 4, Fall 1999, pp. 2158–2162.
[27] B. Liang, Z.J. Haas, Predictive distance-based mobility management for PCS networks, 18th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '99, vol. 3, 21–25 March 1999, pp. 1377–1384.
[28] Z. Lei, C. Rose, Wireless subscriber mobility management using adaptive individual location areas for PCS systems, IEEE International Conference on Communications, ICC 98, vol. 3, 7–11 June 1998, pp. 1390–1394.
[29] Y.-B. Lin, I. Chlamatac, Wireless and Mobile Network Architecture, Wiley, New York, 2001.
[30] Y.-B. Lin, P.-C. Lee, I. Chlamtac, Dynamic periodic location area update in mobile networks, IEEE Transactions on Vehicular Technology, 51(6) November 2002, pp. 1494–1501.
[31] H.-I. Liu, C.-P. Liu, A geography based location management scheme for wireless personal communication systems, IEEE 51st Vehicular Technology Conference, VTC 2000, vol. 2, Spring 2000, pp. 1358–1361.
[32] B.L. Mark, Z.R. Zaidi, Robust mobility tracking for cellular networks, IEEE International Conference on Communications, ICC 2002, vol. 1, 2002, pp. 445–449.
[33] M. McGuire, K.N. Plataniotis, A multi-model filter for mobile terminal location tracking, IEEE 56th Vehicular Technology Conference, VTC 2002, Fall 2002, vol. 2, pp. 1197–1201.
[34] S. Mishra, O.K. Tonguz, Most, Recent interaction area and speed-based intelligent paging in PCS, IEEE 47th Vehicular Technology Conference, vol. 2, 4–7 May 1997, pp. 505–509.
[35] Z. Naor, H. Levy, Minimizing the wireless cost of tracking mobile users: an adaptive threshold scheme, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '98, vol.2, 29 March–2 April 1998, pp. 720–727.
[36] Z. Naor, H. Levy, LATS: a load-adaptive threshold scheme for tracking mobile users, IEEE/ACM Transactions on Networking, vol. 7(6), December 1999, pp. 808–817.
[37] C. Rose, Minimizing the average cost of paging and registration: a timer based method, Wireless Networks 2 (1996) 109–116.
[38] M.A. Spirito, A.G. Mattioli, On the hyperbolic positioning of GSM mobile stations, URSI International Symposium on Signals, Systems, and Electronics, ISSSE 98. 29 September–2 October 1998, pp. 173–177.

[39] R. Subrata, A.Y. Zomaya, Location management in mobile computing, ACS/IEEE International Conference on Computer Systems and Applications 2001, pp. 287–289.
[40] R. Subrata, A.Y. Zomaya, A comparison of three artificial life technique for reporting cell planning in mobile computing, IEEE Trans. on Parallel and Distributed Systems 14 (2) (February 2003) 142–153.
[41] R. Subrata, A.Y. Zomaya, Evolving cellular automata for location management in mobile computing networks, IEEE Transactions on Parallel and Distributed Systems 14 (January 2003) 13–26.
[42] J. Taheri, A.Y. Zomaya, The use of a hopfield neural network in solving the mobility management problem, IEEE/ACS International Conference on Pervasive Services, ICPS 2004, 19–23 July 2004, pp. 141–150.
[43] J. Taheri, A.Y. Zomaya, A genetic algorithm for finding optimal location area configurations for mobility management, The IEEE Conference on Local Computer Networks (LCN), 2005. 30th Anniversary, 15–17 November 2005, pp. 568–577.
[44] J. Taheri, A.Y. Zomaya, A simulated annealing approach for mobile location management, 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS), 04–08 April 2005, pp. 194–194.
[45] J. Taheri, A.Y. Zomaya, Realistic simulations for studying mobility management problems, International Journal of Wireless and Mobile Computing 1 (8) (2005).
[46] O.K. Tonguz, S. Mishra, D.M. Grasso, Impact of random user motion on locating subscribers in mobile networks, IEEE International Conference on Communications, ICC ’99, vol. 3, 1999, pp. 1746–1750.
[47] O.K. Tonguz, S. Mishra, R. Josyula, Intelligent paging in wireless networks: random mobility models and grouping algorithms for locating subscribers, IEEE 49th Vehicular Technology Conference, vol. 2, July 1999, pp. 1177–1181.
[48] W. Wang, I.F. Akyildiz, G.L. Stuber, An optimal partition algorithm for minimization of paging costs, IEEE Global Telecommunications Conference, GLOBECOM ’00, vol. 1, 2000, pp. 188–192.
[49] N. Watanabe, T. Imaizumi, Fuzzy k-means clustering with crisp regions, The 10th IEEE International Conference on Fuzzy Systems, 2001, vol. 1, 2–5 December 2001, pp. 199–202.
[50] Y. Watanabe, M. Yabusaki, Mobility/traffic adaptive location management, IEEE 56th Vehicular Technology Conference, VTC 2002, Fall 2002, vol. 2, 2002, pp. 1011–1015.
[51] V.W.S. Wong, V.C.M. Leung, An adaptive distance based location update algorithm for next generation PCS networks, IEEE Journal on Selected Areas in Communications 19 (2001) 1942–1952.
[52] V.W.-S. Wong, V.C.M. Leung, An adaptive distance-based location update algorithm for PCS networks, IEEE International Conference on Communications, ICC 2001, vol. 7, 2001, pp. 2001–2005.
[53] H.-K. Wu, M.-H. Jin, J.-T. Horng, C.-Y. Ke, Personal paging area design based on mobile’s moving behaviors, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, vol. 1, 2001, pp. 21–30.
[54] M.P. Wylie-Green, P. Wang, GSM mobile positioning simulator, IEEE Emerging Technologies Symposium: Broadband, Wireless Internet Access, 2000, 5pp.
[55] Z. Yang, X. Wang, Joint mobility tracking and hard handoff in cellular networks via sequential Monte Carlo filtering, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, vol. 2, 2002, pp. 968–975.
[56] A. Wing Ho, W. Yuen, S. Wong, A dynamic location area assignment algorithm for mobile cellular systems, EEE International Conference on Communications, ICC 98, vol. 3, 7–11 June 1998, 1385–1389.
[57] Q-A.Zeng, D.P. Agrawal, Performance analysis of a handoff scheme in integrated voice/data wireless networks, Proceedings of the IEEE VTC-00, September 2000, pp. 845–851.

Javid Taheri was born in 1975 in Iran. He received his Bachelor of Science in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 1998. He also received his Masters degree from the same university in the field of Intelligent Control and Robotics in 2000. He is currently a Ph.D. student at the School of Information Technologies, The University of Sydney, Australia specialising in the field of Mobile Computing. Mr. Taheri published extensively in the field of robotics. Before commencing his Ph.D. studies he worked as a professional software developer for the largest car manufacturer in Iran, IKCO. His main duty was to develop sophisticated programs for machine vision applications, such as, car assembly lines and body shops. Currently, he is focusing on applying intelligent algorithms to target the mobility management problem.

Albert Y. Zomaya is currently the Head of School and *CISCO Systems Chair Professor of Internetworking* in the School of Information Technologies, The University of Sydney. Prior to that he was a Full Professor in the Electrical and Electronic Engineering Department at the University of Western Australia, where he also led the Parallel Computing Research Laboratory during the period 1990–2002. He served as Associate-, Deputy-, and Acting-Head in the same department, and held visiting positions at Waterloo University and the University of Missouri-Rolla. He is the author/co-author of 6 books, 270 publications in technical journals and conferences, and the editor of 7 books and 7 conference volumes. He is currently an associate editor for 16 journals, the Founding Editor of the *Wiley Book Series on Parallel and Distributed Computing*, and the Co-editor of the *Wiley Book Series on Bioinformatics*, and the Editor-in-Chief of the *Parallel and Distributed Computing Handbook* (McGraw-Hill, 1996). Professor Zomaya was the Chair the *IEEE Technical Committee on Parallel Processing* (1999–2003) and currently serves on its executive committee. He has been actively involved in the organization of national and international conferences. He received the *1997 Edgeworth David Medal* from the Royal Society of New South Wales for outstanding contributions to Australian Science. In September 2000 he was awarded the *IEEE Computer Society's Meritorious Service Award*. Professor Zomaya is a chartered engineer (CEng), a Fellow of the AAAS, IEEE, and the IEE (UK), and a Distinguished Engineer of the ACM. His research interests are in the areas of high performance computing, parallel algorithms, networking, and bioinformatics.