An Efficient Mobility Management Strategy for Personal Communication Systems

Yigal Bejerano* and **Israel Cidon†**
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel

ABSTRACT

Personal Communication Systems (PCS) enable people to communicate independent of their location. For tracking the location of mobile users the system must maintain a *Location Management* mechanism, which maps user addresses to their current location. The increasing population of mobile users leads to congestion problems in these systems, and motivates the development of more efficient management schemes. This work presents a novel hierarchical Location Management scheme, in which every level of the hierarchy represents a partition to geographic regions. Within each level of the hierarchy the system records the location of every mobile user to a certain degree of accuracy. The degree of accuracy is increased as we go down the levels until we reach the node to which the mobile user is attached. We develop distributed procedures for locating the mobile users (termed the Search operation) and updating the system location records (termed the Update operation) with user movements. The proposed scheme guarantees upper bound on the procedures costs: The amortized complexity of the mobile user update operations is $O(Move \cdot \log Move)$, where $Move$ is the total geographic distance that the mobile user has traveled. The upper bound of a search operation is linear with the distance between the search originator and the target node. These upper bounds do not depend on the network size. Therefore, the proposed scheme is attractive for the next generation of PCS. The management system is also suitable for supporting anycast and territory restricted users.

Keywords: Wireless Systems, Personal Communication System, Cellular Systems, Mobility Management, Location Management, Anycast.

1 Introduction

Personal Communication System (PCS) enables people to communicate independent of their location. In contrast to the telephone number in traditional telecommunication systems that specifies the location of the end user, the PCS subscriber number does not provide the location of the mobile user. Therefore, the system must maintain a *Location Management* mechanism, also termed *Mobility Management*, for locating it's mobile users. This mechanism maps subscriber numbers to the current location of the requested users. A Location Management mechanism is composed of three components: A *system database* for mapping subscriber numbers to locations, an *update* operation for informing the system database regarding changes in mobile user locations, and a *search* operation for locating the mobile users based on the information stored at the system database.

Different location management methods are described in the literature. These methods differ in the way the location database is organized, which also affects the update and search algorithms. Several criteria are used for evaluating the efficiency of the various methods. Such as their communication cost, the time to perform these operations, the processing overhead associated with them, the amount of memory required, etc. In this work we mainly focus on the communication cost of the update and search operations.

The current PCS, such as GSM [12],IS-41 [1], Mobile-IP [9], use similar schemes for mobility management. Each mobile user is associate with an *home location server*, that points to the current location of the mobile user. While these schemes are widely used, the fast growth of the mobile user population leads to congestion problems in large systems [14, 16]. This motivates the development of new Location Management methods with better scalability. The newly proposed methods are usually based on the local behavior of mobile users, i.e., users tend to communicate with mobile users that are relatively close to their location and mobile users move between adjacent areas.

The new proposals can be divided into two groups. The first includes incremental improvements to the current home location server scheme such as caching strategies [15, 11], pointer forwarding techniques [10], local anchoring scheme [2], etc., summarized in the surveys [1, 6, 14]. The second group is based on an hierarchical organization of a distributed system database for locating mobile users. Most of these approaches are based on the organization of the location servers in a tree structure (see [16, 17, 8, 7, 5, 13]). A different hierarchical scheme has been proposed by Awerbuch and Peleg [3, 4]. They use a graph-theoretic concept of *regional matching* for implementing an efficient tracking mechanism. Their work is especially important since, to the best of our knowledge, it is the only one that guarantees upper bounds for the communication cost of the up-

*This work was partially supported by Intel Israel.
†also with SUN Microsystems Labs, CA.

date and search operations relative to the lower bound of these operations. Consider a communication network with N nodes and M mobile stations and a parameter δ that represents the logarithm of the graph diameter. Then, the upper bound for the update operations is $O((\delta \cdot log^2 N + \delta^2/logN) \cdot logM \cdot Move)$, where $Move$ is the total distance that the mobile user has traveled measured in communication metric. The upper bound for the search operation is $O(log^2 N \cdot \log M \cdot Dist)$, where $Dist$ is the communication distance between the search originator and the target node.

In this work we present a new hierarchical method that employs some of the principles used by Awerbuch and Peleg [4]. Our scheme distinguishes between the communication network and the coverage graph. Both graphs are composed of the same set of nodes but represent distances of different metrics. The first represents the communication cost between the location servers, while the second represents the geographic distance between regions. Each node represents both a location server and a geographic environment in which mobile users may travel. We assume that the communication network and the coverage graph are correlated, such that the communication cost between each pair of nodes is related to their geographic distance. The proposed scheme has the following upper bounds. The total cost of a mobile user update operations is $O(Move \cdot \log Move)$. The cost of a search operation is linear to the distance between the search originator and the target node. In contrast to [4], these upper bounds do not depend neither on the network diameter or the number of nodes. This makes the proposed scheme attractive for large PCS networks.

This paper is organized as follows. Section 2 describes the system model. Section 3 presents the principles of the proposed scheme. Section 4 contains an analysis and upper bounds for the communication cost. Section 5 explains the partition methods used by our scheme, and section 6 presents further additional advantages of the scheme.

2 The Model

In the sequel we consider the infrastructure communication network, termed the *infrastructure graph.* The infrastructure graph consists of a set of nodes V. Each node manages the operations of several base stations and maintains a list of all the mobile users that are attached to these base stations. Around each node $v \in V$ we define a vicinity, and we assume that node v is located at its center. This vicinity represents some logical managed area. Furthermore, the union of all the vicinities defines the *coverage area* of the system. Hence, each node induces a geographic vicinity and it serves as a location server of the PCS. Two nodes are called *adjacent* if their vicinities meet. Mobile user may freely move in the coverage area. We assume that a local movement assumption is sustained, i.e. user may move from its current vicinity only to adjacent ones. Such a transition between two vicinities is also represented as a move between the corresponding nodes. Our basic distance unit in a given system is the radius of the smallest vicinity. Hence the distance between any pair of nodes is at least 1.

The nodes of the infrastructure graph are connected by communication links that are represented by the edges of this graph. We assume that the communication system is reliable and includes an efficient routing mechanism. As a result, each message that is sent from a source node to a destination node travels through the shortest path between the nodes, and reaches its destination. The above model combines two metrics: A geographic distance and a communication cost. On one hand, the mobile users are physically traveling within the coverage area, measured by a geographic distance metric. On the other hand, the management operations are performed over the communication network. Therefore the model is required to represent both metrics.

We compute from the infrastructure graph a complete graph $G(V, E)$, where each edge $(u, v) \in E$ has two weights c and d. c is the cost of the shortest communication path between nodes v and u. d is the geographic distance between these nodes. In principle, there is no necessary relation between the two metrics. The communication cost between two adjacent nodes may be very high, and vice versa. In this work we assume that in practice this is not the case, and the communication cost is related to the distance. We assume the existence of a constant $\alpha > 0$, such that for each edge $(u, v) \in E$ with weights c and d $c \leq \alpha \cdot d$. α is termed the *correlation* constant.

3 The Principles of the Proposed Scheme

3.1 The Hierarchical Organization Of The System

The proposed system is based on an hierarchical approach in which every level of the hierarchy represents a partitioning to geographic regions. The hierarchy contains $L + 1$ levels numbered in increasing order from 0 to L. L is at most logarithmic with the geographic diameter of the graph. Each level of the hierarchy records the location of every mobile user to a certain degree of accuracy. As we move from the higher levels to the lower levels the degree of accuracy is increased , until we reach the node to which the mobile user is attached.

At each level $k \geq 1$ the coverage area is divided to continuous geographical regions with one or more nodes. Adjacent regions may overlap each other. Two regions at the same level are called *overlapping* if there is a non-empty group of nodes that is included at both regions. Each region is composed of two sub-regions. The *core* is the center of the region and its radius is at least 2^k. It is surrounded by an area with a width of at least 2^k that is termed the *periphery.* The set of all cores at any given level k covers all the nodes without overlapping each other. However, adjacent regions are overlapping as a result of wrapping each core with a periphery.

As a result, at each level k, every node may be included in several regions but it is included at exactly one core. An example for such geographic partition is depicted in Figure 1. This figure shows a partition of the coverage area into core regions at a given level k. One of these core is surrounded by the region periphery of width 2^k. At level 0 we use a different structure. Each node is considered as a separate region and therefore the regions don't overlap. Level 0 is used for the final location of a required mobile user. The regions of each level increase with the level index. Starting from level 0, in which each node defines a region of its own, up to level L, where all nodes are included in a single region. Such a construction is presented in Figure 2, for the case of a uni-dimensional network.

3.2 The System Database

We turn to describe the principles of the system database and how it records the location of all the mobile users. Similar to [4], the system maintains a distributed database that records the location of every mobile user at several control

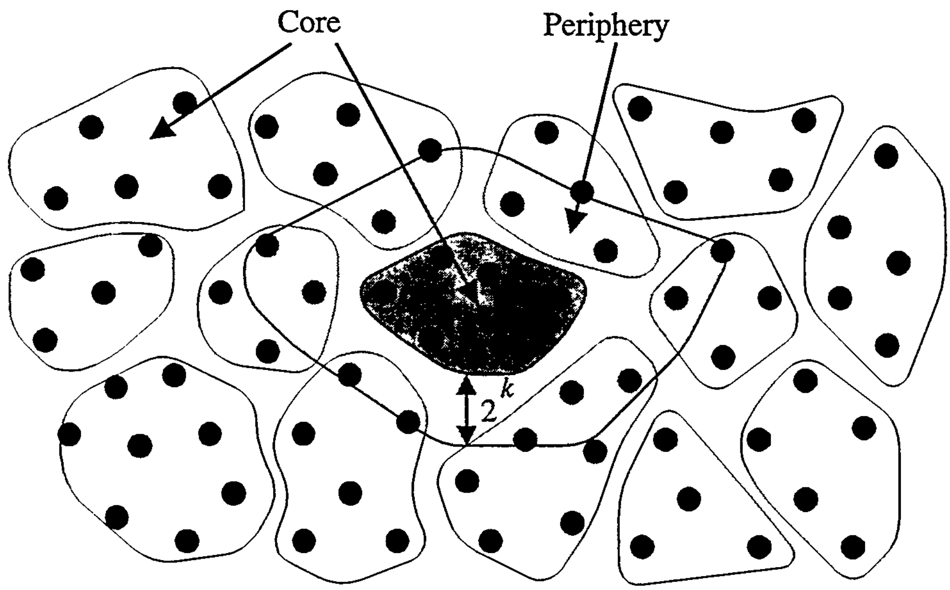

Figure 1: An example of a geographic partition of the coverage area to regions at level k.

nodes with a different degree of accuracy. At each level, every region has a single node that holds a list of mobile users located at this region at both the core and the periphery (the list does not necessarily include all the mobile users in the region). We refer to this node as the *control node* of the region and the region is considered as the *jurisdiction region* of this control node. The mobile users list is updated by update operations, and it is used for locating mobile users in this region at search operations.

At each level of the hierarchy, every mobile user is recognized at a single control node in whose jurisdiction region he is located. A control node at level k that recognizes a requested mobile user points to the region at level $k-1$ where this user resides. This is done by pointing to the control node of the corresponding region at level $k-1$. Hence, each mobile user has a *pointer path*, starting from a single control node at the highest level (level L), passing through all the hierarchy levels, and ending at the node at level 0 to which the mobile user is attached. Consequently, a search operation for a mobile user reaches a control node within its pointer path and follows the path until reaching the mobile user. The pointer paths of two mobile users that are attached to the same node but have different pointer paths, is illustrated in Figure 2. This difference results from different routes that the mobile users followed before they reached this node.

As a result of the hierarchy construction and the database organization, the scheme possesses the following two properties. First, let k be the lowest integer such that the distance between a search originator and the destination is less or equal 2^k, then the search operation queries control nodes at most at level k. Second, updating the location of a mobile user at level $k+1$ is done only if the mobile user has traveled at least a distance of 2^k since the last time this level has been updated. Consequently, update and search operations incur a communication cost that is proportional to the geographical distance between the source and the destination nodes.

3.3 The Search and The Update Operations

In the following we describe the way the system database is updated regarding the current location of a mobile user as it moves between areas. An update operation is performed when a mobile user leaves a region at any level k and moves to another region at this level. For simplicity, consider a mobile user that moves from region A_k to an adjacent region B_k at a specific level k. According to our definition, a mobile user leaves region A_k only if it moves from a node included in the periphery of region A_k to a node outside that region. The mobile user enters region B_k if the new node is included at the core of region B_k. As a result of this movement, the mobile user's pointer path must be updated by replacing the control node of region A_k with the control node of region B_k. This operation is done by updating the control node of level $k+1$ to point to the control node of region B_k (instead of that of region A_k), and the latter is set to point to the tail of the path. Level $k+1$ will be updated again only if the mobile user leaves region B_k. At this point, the mobile user is located inside the core of region B_k. According to our construction, its distance from this region boundary is at least 2^k, as the width of its region periphery. Therefore, the mobile user has to travel at least a distance of 2^k before it leaves region B_k, and only then an additional update operation of level $k+1$ will take place.

From the above discussion, it is clear that at any level, a control node is not necessarily familiar with all the mobile users that are located within its jurisdiction region, or even at its core. This is because the coverage area is divided at each level to overlapping regions and the way the update operation works. However, each mobile user is recognized at one of the control nodes in whose jurisdiction region it resides at level k.

We turn to describe the principles of the search opera-

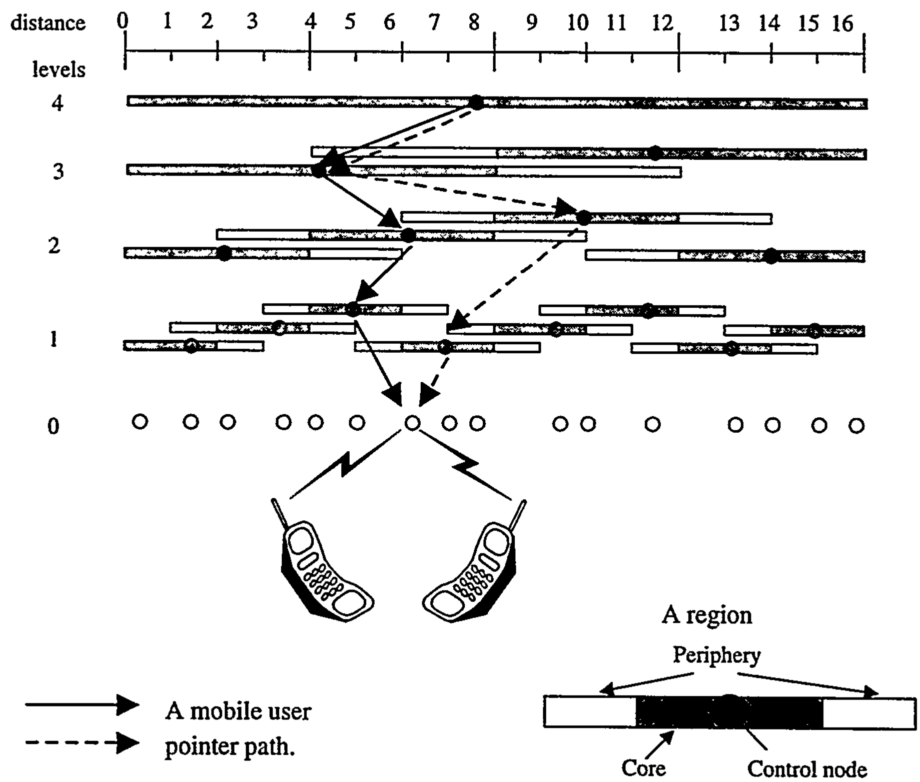

Figure 2: An example of an hierarchical system with two pointer paths.

tion. The search operation is similar to the find operation at [4]. This operation is composed of two parts: Finding a control node that is included in the pointer path of the required mobile user, and then following this path until the mobile user is reached. Notice that all the pointer paths start at the single control node of the highest level (level L). Hence, it is always possible to access this node and from this point track the entire pointer path. Although this is a plausible solution, its cost may be high in comparison with the communication cost of the shortest path between the source and destination nodes. Our goal is to locate relevant mobile users using low communication overhead in comparison with the communication cost over the shortest path. We achieve this goal by finding a control node in the user's pointer path at the lowest possible level.

For clarifying the search operation, suppose that a source node s looks for a mobile user ω, which is attached to a destination node d. Node s searches ω at increasing distances. First, it checks whether ω is directly attached to it, then it checks whether the mobile user is at distance of no more then 2^1, 2^2, ..., and so on, until it is found. For this search we define for each node a set of localities. We term the *k-locality* of node s as a circular territory of radius 2^k around node s. Regard that this locality contains all the nodes that their distance from s is at most 2^k and it is entirely included at the region at level k whose core contains node s, see Figure 3. Node s examines if ω is located in its k-locality by querying the databases at all the control nodes at level k that their jurisdiction region overlap this territory. If ω is located at the k-locality of s, then one of these control nodes is included at the pointer path of ω at level k or less. This path leads to the destination node d.

4 The Complexity of the Proposed Scheme

We use the following characteristics for the system analysis. Let the infrastructure graph be a graph with α correlation constant, N nodes and M users. The system hierarchy contains $L+1$ levels, where L is at most the geographic diameter of the system coverage area. The identity of each node or mobile user is represented by $log(N)$ or $log(M)$ bits correspondingly. We assume that M is much higher then N. Therefore, we ignore the amount of memory that is required for holding the configuration information, and only consider the total amount of memory that is required for maintaining the mobile user pointer paths. We also ignore the communication cost that is required for setting the initial pointer path of a mobile user, since this operation is performed only once. Now consider the partition of the coverage area into regions at each level of the hierarchy. Let the *radius* of a region be the distance from the region's control node to the most distant node in this region. We require that the geographic radius of a region at any level k is upper bounded by $R \cdot 2^k$ of a given constant R. R is termed the *stretch factor* of the partition. We also require that each k-locality has at most T overlapping regions at level k. T is called the *overlap factor* of the partition. In section 5 we present particular partition methods with small R and T.

Initially, we turn to calculate the amortize complexity of update operations. Consider a mobile user ω that has moved

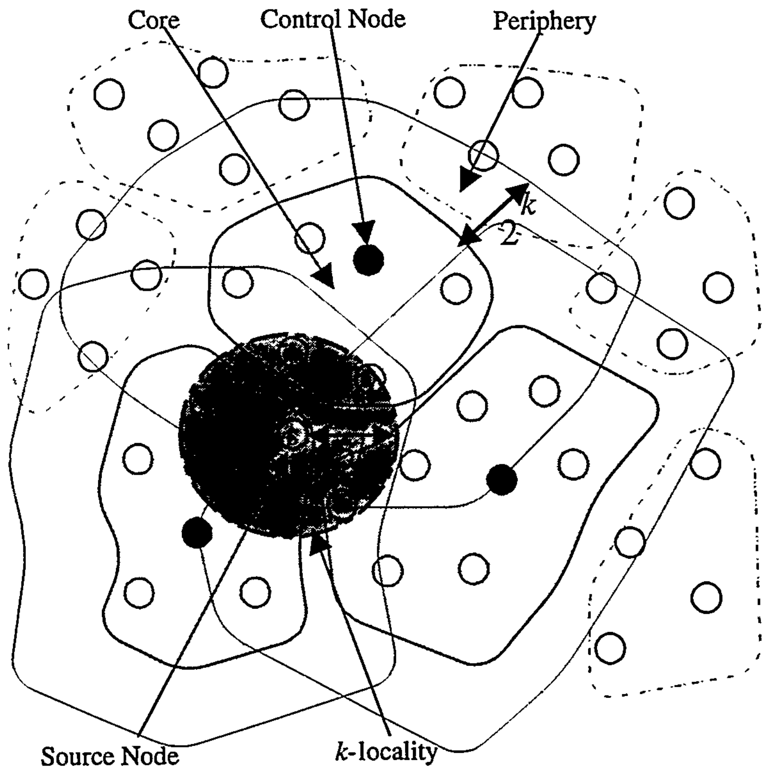

Figure 3: An example of a k-locality of a node that overlaps three regions.

a total geographic distance $Move$ since its activation. In its way it triggers a sequence of update operations. First, consider a single movement of ω, in which it has moved from region A_k to an adjacent region B_k at some level k. As a result, the pointer path of ω is modified at levels 0 up to level $k+1$. During the update operation an UPDATE message propagates upward the hierarchy levels, modifying in its way the user pointer path, until reaching level k+1. Then a RELEASE message is transmitted downward for releasing unnecessary resources. The UPDATE and RELEASE messages carry the identity of the mobile user and the changes in the pointer path. Since $M >> N$, the messages length is $O(\log M)$. Moreover, the geographic distance between two successive nodes in a pointer path at levels $j+1$ and j is at most $(R \cdot 2^j + R \cdot 2^{j+1})$. Therefore, the communication cost of these messages is bounded by following:

$$2 \cdot \alpha \cdot \log M \cdot \sum_{j=0}^{k}(R \cdot 2^j + R \cdot 2^{j+1}) =$$

$$6 \cdot \alpha \cdot R \cdot \log M \cdot \sum_{j=0}^{k} 2^k \leq$$

$$12 \cdot \alpha \cdot R \cdot \log M \cdot 2^k$$

As a consequence, the communication cost of this update operation is $O(\alpha \cdot R \cdot \log M \cdot 2^k)$.

Now, let consider the maximum number of times that the mobile user may pass from one region to an adjacent region at any level of the hierarchy. According to the region partition, the mobile user has to travel at least a distance of 2^k between two successive transitions of regions at some level k. Therefore, the number of region transitions at level k is bounded by $\lfloor Move/2^k \rfloor$. The communication cost of the entire update operations is bounded by $\sum_{k=0}^{\lfloor \log Move \rfloor} \lfloor Move/2^k \rfloor \cdot O(\alpha \cdot R \cdot 2^k \cdot \log M)$. Hence, the amortized complexity of the update operations is $O(\alpha \cdot R \cdot \log M \cdot Move \cdot \log Move)$.

Let us now turn to the search operation in which the geographic distance between the search originator and the destination node is $Dist$, and let k be $\lceil \log Dist \rceil$. The mobile user is located at the k-locality of the search originator. Therefore, only control nodes at level k and below are queried in this operation.

The search operation works as follows. The search originator checks its localities set in increasing order until it reaches the k-locality. At each iteration j, $j \leq k$, the originating node queries the control nodes at level j that their jurisdiction region overlap its j-locality, whether one of them is included in the user pointer path. When such a node is found, a trace message follows the user pointer path until reaching the destination node. In our calculation we use the following. First, all the messages in this operation are of size $O(\log M)$. Second, each j-locality of every node overlaps at most T regions at level j. Moreover, the geographic distance between the search originator to a control node of an overlapping region at level j is at most $(R+1) \cdot 2^j$. Third, the geographic distance between two successive nodes at level j and $j-1$ in the trace message routes is at most $(R \cdot 2^j + R \cdot 2^{j-1})$. As a result, the communication cost of the search operation is bounded by

$$\sum_{j=1}^{\lceil \log Dist \rceil} \left\{ O(\alpha \cdot R \cdot \log M \cdot (2^{j-1} + 2^j)) + O(\alpha \cdot R \cdot T \cdot \log M \cdot 2^j) \right\} =$$

$$\sum_{j=0}^{\lceil \log Dist \rceil} O(\alpha \cdot R \cdot \log M \cdot T \cdot 2^j) =$$

$$O(\alpha \cdot R \cdot T \cdot \log M \cdot Dist)$$

Hence, the communication cost of a search operation is $O(\alpha \cdot R \cdot T \cdot \log M \cdot Dist)$.

We have shown above that the upper bounds of the search [1] and update operations depend on the distance between and source and the destination node as well as the correlation constant α, the partition stretch factor R and the partition overlap factor T. In contrast to [4], these bounds do not depend on the system size, the size of the coverage area, the number of nodes in the infrastructure graph or the number of mobile users in the system. Therefore, the scheme is scalable and is suitable for large systems.

Finally, we examine the system memory requirements. The proposed scheme is required to maintain a pointer path for each mobile user. Since the system consists of $L+1$ levels, the size of every pointer path is exactly L, where each entry in this path includes a mobile user and a node identities. Therefore, the total memory requirement is $O(M \cdot L \cdot \log M)$.

5 Partition Methods

We have shown that the way every level is divided into regions affects the cost of the update and search operations. Each partition method is characterized by two parameters: the stretch factor R and the overlap factor T. The goal of our partition methods is to minimize these parameters and the product $T \cdot R$. In the following we present two simple partition methods in which the coverage area is divided into hexagons at each level of the hierarchy. Although the two methods seem similar they are derived from different observations and represent different trade-offs. In the sequel, we refer to a level k of the hierarchy and show the way it is divided into regions. For each method, we present the partitioning of the coverage area into region cores. Each core is surrounded by a periphery of width 2^k. For simplicity we assume that the control node of a region is located at its center.

The first method objective is to reduce the stretch factor R of the partition. This is the only partition characteristic that affects the cost of the update operations. Hence, by reducing this parameter the cost of the update operations is decreased. Such a partition is appropriate for systems where the number of update operations dominates the number of search operations. In this partition method, the core shapes are the smallest hexagon that circumscribes a circle with radius 2^k, as described at figure 4. The radius of a core is $2^k/\cos 30 = (2/\sqrt{3}) \cdot 2^k$. Therefore, the stretch factor $R = (1 + 2/\sqrt{3}) \cdot 2^k = 2.15 \cdot 2^k$, the overlap factor $T = 7$ and the product $R \cdot T = 15.08$. Note that choosing a smaller hexagon increases dramatically the overlap factor, since two regions may overlap even if their cores are not adjacent.

The second method attempts to minimize the number of accesses to control nodes during the search operations. This goal is achieved by reducing the overlap factor T. Here, a core shape is an hexagon whose sizes are of length $4 \cdot 2^k$, as described at figure 5. This is also the radius of the core, and the stretch factor is $R = 5 \cdot 2^k$. The minimal distance between two regions, that their cores are not adjacent, is at least $2 \cdot 2^k$. As a result, at most 3 regions may overlap any k-locality, and the overlap factor is 3, which is the minimal possible value. The product $R \cdot T = 15$. This partition method is suitable for systems with a high rate of search operations compared to update operations.

Next, we present a method that generates in sequence all the hierarchy levels. We use the fact that the core radius at level $k+1$ is twice then the radius of a core at level k. First, find an hexagon of radius $R \cdot 2^L$ that contains all the coverage area. This hexagon defines the highest level of the hierarchy - level L. The other levels are defined recursively from level L down to level 1. The recursion step is demonstrated in figure 6. Consider that the region cores at level $k+1$ are defined. The radius of these cores is $(R-1) \cdot 2^{k+1}$. Now let define the region cores of level k. For each core at level $k+1$, generate an hexagon of radius $(R-1) \cdot 2^k$ with the same center. Then, surround every new generated hexagon with six hexagons of the same size. This step defines all the cores at level k and enables us to define the regions of all the levels of the hierarchy.

6 Extensions and Additional Features

The proposed scheme offers several more advantages that make it attractive for PCS. Some of them are summarized in the following.

Our scheme allows separate update and search regions. This separation enables to design different update and search regions based on different considerations (geometric shape, movement probability, communication cost, etc.), such that the entire system overhead is decreased. Moreover, the region design can be adaptive to system state and their shapes may change dynamically in response to events like the time and the date, changes in the communication network and in the movement pattern of the mobile users.

Another useful application of our search/update scheme is a *mobile anycast service*. Many times users are looking for an available service and not necessarily a specific mobile user. For instance a close by policeman or a free taxi cab. Assume that the mobile users that provide a specific service share an *anycast address*, and the system should find a close by service provider to the search originator. Our proposed scheme can efficiently find a close by provider. This operation is performed as follows: The scheme starts a search operation until it finds a proper mobile user, let say at level k. The resulted mobile user isn't necessary the closet one, but its distance from the source node is no more then $(R+1) \cdot 2^k$, where R is the partition stretch factor. If it is required to locate the closest service provider, then the scheme checks whether there is a closer provider by checking the m-locality of the source node, where $m = k + \lceil \log_2(R+1) \rceil$. If there is a closer service provider then it will be found.

Finally, our system can also efficiently support *territory restricted users*. These restricted users can receive communication services only in a restricted territory, like a city, a

[1]For users, which are frequently searched by others, we can further reduce the cost of the search operations by making these users recorded at more then a single control node at each level.

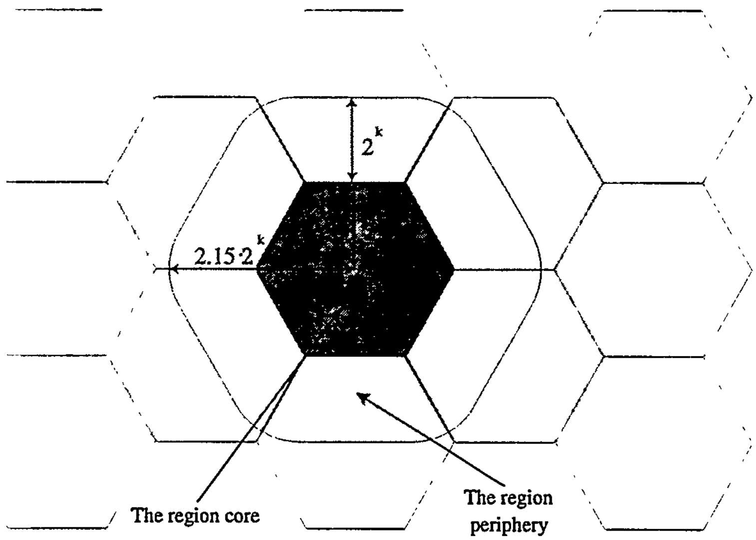

Figure 4: The first partition method with $R = 2.15$ and $T = 7$.

county or a state. The proposed scheme used this restriction for reducing the overhead of update and search operations where such mobile users are involved. Consider a mobile user that is allowed to move in a restricted territory with a geographic radius of at most 2^k. This territory is entirely included in a single region at level k and at any higher level, independent of the partition method. Hence, the pointer path of the mobile user at level k and above becomes constant, simply by passing it thought the control nodes of these regions. This assignment also enables to locate the mobile user at level k at most.

References

[1] F. Akyildiz and J. S. M. Ho. On Location Management For Personal Communications Networks. *IEEE communication magazine*, Vol 34 No 9. Sep. 1996.

[2] F. Akyildiz and J. S. M. Ho. Dynamic Mobile User Location Update For Wireless PCS Networks. *ACM-Baltzer Journal of Wireless Networks*, Vol. 1, No. 2, July 1995, pp 187-196.

[3] B. Awerbuch and D. Peleg. Sparse Partitions. *31st IEEE symposium on Foundation of Computer Science* Oct. 1990, pp 503-513.

[4] B. Awerbuch and D. Peleg. Concurrent Online Tracking of Mobile Users. *SIGCOM symposium on Communication, Architectures and Protocols*, Oct 1991.

[5] B. R. Badrinath, T. Imielinski and A. Virmani. Location Strategies For Personal Communication Network *Proc. Workshop on Networking of Personal Comm. Applications*, Dec. 1992.

[6] P. Bhagwat, C. Perkins and S. Tripathi. Network Layer Mobility : An Architecture and Survey. *IEEE personal communication*, June 1996.

[7] S. Dolev, D. K. Pardhan and J. L. Welch. Modified Tree Structure For Location Management in Mobile Environments *Proc. IEEE INFOCOM 1995.*

[8] C. Eynard, M. Lenti, A. Lombardo, O. Merengo, and S. Palazzo. A Methodology For Performance Evaluation of Data Query Strategies in Universal Mobile Telecommunication Systems (UMTS). *IEEE Journal on Selected Areas in Communications*, Vol 13, No 5, June 1995.

[9] C. Huitema. Routing in the Internet. *Printice-Hall*, 1995.

[10] R. Jain, Y. B. Lin, C. Lo and S. Mohan. A Forwarding Strategy To Reduce Network Impacts of PCS. *Proc. IEEE INFOCOM 1995.*

[11] R. Jain, Y. B. Lin and S. Mohan. A Caching Strategy To Reduce Network Inpacks of PCS. *IEEE Journal on Selected Areas in Communications*, Vol 12, No 8, Oct. 1994.

[12] M. Mouly and M. B. Pautet. The GSM System For Mobile Communication. M. Mouly, 49 rue Louise Bruneau, Palaiseau, France 1992.

[13] M. van Steen, F. J. Hauck, P. Homburg and A. S. Tanenbaum. Locating Objects in Wide-Area Systems. *IEEE communication magazine*, Vol 36 No 1.Jan 1998.

[14] S. Tabbane. Location Management Methods For Third-Generation Mobile Systems. *IEEE communication magazine*, Vol 35 No 8. Aug. 1997.

[15] F. Teraoka, Y. Yokote and M. Tokoro. A Network Architecture Providing Host Migration Transparency. *Proc. of ACM SIGCOM*, pp 209-220, Sep. 1991.

[16] J. Z. Wang. A Fully Distributed Location Registration Strategy For Universal Personal Communication Systems. *IEEE Journal on Selected Areas in Communications*, Vol 11, No 6, Aug 1993.

[17] J. Weinmiller. Survey on Location and Packet Forwarding Strategies in Networks With Mobile Hosts. URL: http://www-tkn.ee.tu-berlin.de/~jost/pub.html . June 1994.

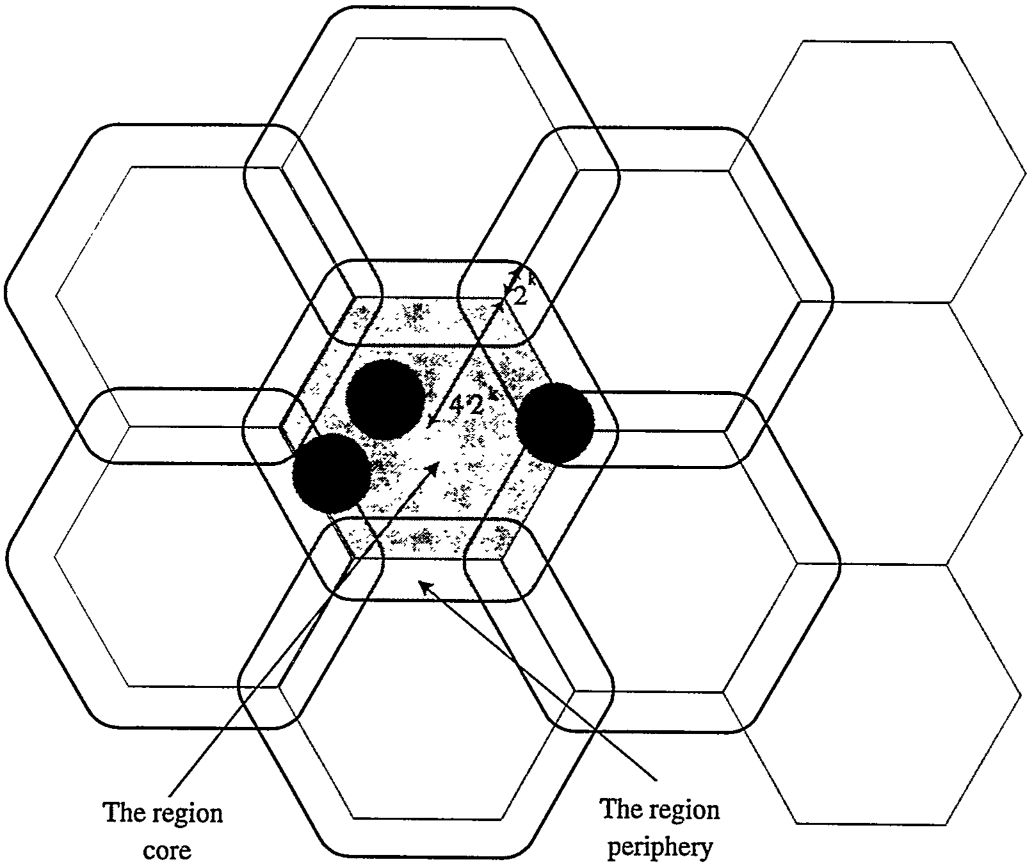

Figure 5: The second partition method with $R = 5$ and $T = 3$.

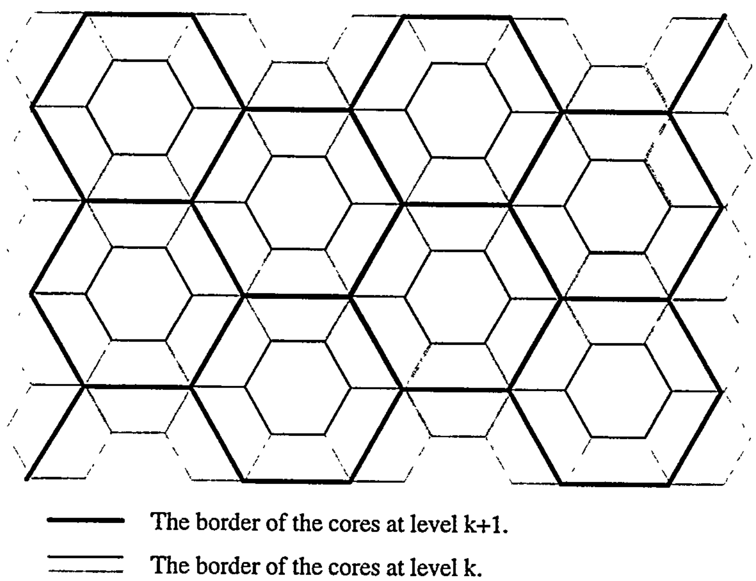

Figure 6: Defining the region cores at level k based on the partition of level $k + 1$.