Wireless Networks 9, 409–420, 2003
© 2003 Kluwer Academic Publishers. Manufactured in The Netherlands.

An Anchor Chain Scheme for IP Mobility Management

YIGAL BEJERANO *,**
Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974, USA

ISRAEL CIDON
Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Abstract. This work presents a simple mobility scheme for IP-based networks, termed the "anchor chain" scheme. The scheme combines pointer forwarding and caching methods. Every mobile host (MH) is associated with a chain of anchors that connects it to its home agent. Each anchor defines the location of the MH at a certain degree of accuracy. The accuracy is increased along the chain until the attachment point of the MH is reached. We develop distributed procedures for updating the anchor chain (binding operation) with MH movements and for delivering messages to a MH (delivery operation). In terms of worst case performance, the total cost of the binding operations is $O(\mathit{Move} \log \mathit{Move})$, where *Move* is the total geographic distance that the MH has traveled since its activation. The total length of the MH's pointer path is linear with the distance between the MH and its home network, and the delivery cost is near optimal. In addition, the anchor chain of a MH is determined dynamically with no need for preliminary definitions of static anchors or regions. Our simulation results show that the anchor chain scheme also yields lower average overheads for both the binding and the delivery operations than other methods that are described in the literature, including the current home approach. We believe that the proposed scheme is scalable, fairly easy to implement and there fore attractive for supporting MHs.

Keywords: wireless systems, mobile-IP, personal communication system, cellular systems, mobility management, location management

1. Introduction

In recent years we have been facing a rapid growth in the need to support mobile hosts over global networks. Originally, the TCP/IP protocol suite was not designed to support mobile hosts. Typically, each host is assigned a unique address that also provides routing information to this host assuming its location is fixed. Therefore, the IP protocol should be augmented with a *mobility management* mechanism for delivering datagrams to their destination hosts independent of their locations. A mobility mechanism is composed of three components: a *location database* for mapping network addresses to locations; a *binding* (*update*) operation for informing the location database regarding changes in mobile host locations; and a *delivery* (*search*) operation for delivering packets to their designated mobile hosts, based on the information stored at the location database. The efficiency of such a mechanism depends on the ability to guarantee delivery of datagrams to their destinations with low overhead of the binding and delivery operations. In this work we present a new simple scheme for supporting mobile users in datagram-based networks from the infrastructure network point of view. The scheme is the first to guarantee a near optimal delivery operation with low binding overhead using worst case analysis. Moreover, simulation results show that also in the average terms, this scheme yields lower overhead relative to other methods that are described in the literature [6,7,9,12–14].

* Corresponding author.
** While the author was at the Electrical Engineering Department at the Technion. Research supported by Eshkol Fellowship from the Israeli Ministry of Science.

The current standard for supporting mobile hosts [9] is based on the home location server approach. Each mobile host (MH) is associated with a *home network* and assigned a permanent IP address based on its home network identifier. As long as it is attached to its home network, packets are delivered to it similar to any wired host. The home network is also required to include a *home agent* that tracks the current location of the mobile host. When the mobile host moves to a new location it acquires a temporary address that defines its new location, termed a *care-off-address* (COA). Then, it updates its home agent about its new COA. When a *correspondent node* (CN) sends a packet to a mobile host, the packet is routed to its home network. If the MH is not attached to its home network, the home agent intercepts the packet and tunnels (see [10]) it to the MH using its COA. In the opposite direction, packets are routed directly to the CN.

Although this mobility management scheme is simple and scalable, it has two main deficiencies. First, packets that are designated to a mobile host are routed via sub-optimal paths [11,13]. Second, the mobile host needs to update its home agent about every movement, even when it is far away from its home network. This makes the update operation expensive and sometimes even impossible when the user movements are too frequent [12]. Different methods for overcoming these problems were described in the literature. Methods for solving the sub-optimal routing problem are described in [13,15]. These methods are based on caching the current location of the mobile host at the correspondent hosts [13] or at the routers [15] for providing near-optimal packet routing to the mobile host. The problem with these approaches is the need to maintain the caches updated. The mobile hosts can handle multiple communication tasks simultaneously, with diverse remote correspondent hosts. Hence, updating the caches of the correspondent hosts may be a very expensive task.

Most of the methods for reducing the binding overhead are based on adding an intermediate entity between the mobile host and its home agent. It enables the mobile host to update a close mobility agent instead of its home agent, in case it is far from its home network. This can be implemented in various ways such as using pointer forwarding techniques [8], local anchoring schemes [5,7], or hierarchical organization of mobility agents as described in [1,2,12,14,16]. These methods usually reduce the cost of binding on behalf of delivering. Some works [5,7] propose to combine caching with local anchoring techniques for reducing both overheads. These works present a probabilistic analysis for the expected cost of the binding and delivery operation. However, they cannot guarantee that in the *worst case* they achieve lower overhead than the current home location server approach for both the binding and delivery operations.

In this work we present a simple mobility scheme, that combines pointer forwarding and caching methods for achieving low cost binding and delivery operations.[1] Since a mobile host is physical device, it usually (but not necessarily) moves from its current vicinity to an adjacent one. Under this local movement assumption the proposed scheme reduces the cost of binding by using a chain of anchors that connects the mobile host to its home agent. Each anchor defines the location of the mobile host at a certain degree of accuracy. These anchors are also used for efficient delivery of datagrams to the MH. Upon initializing a communication with a CN, the MH provides it with a record of its anchor chain. When the CN wishes to send a message to the MH, it selects a node from this record, termed an *access point*, and sends the message to it. From that node the message is forwarded along the chain until it reaches the MH. The scheme guarantees that the selected access point is included in a chain of pointer that tracks the MH location. It enables selection of an access point that is close to the MH current location without sending the message first to the MH home agent.

Under the assumptions of a correlation between communication cost and geographic distances, our scheme guarantees that the (worst case) total cost of a binding operation sequence is $O(Move \log Move)$, where $Move$ is the total geographic distance that the MH has traveled since its activation. The total length of the MH's pointer path is linear with the distance between the mobile host and its home network, and the delivery cost is near optimal. Our simulation results show that also in the average terms the anchor chain scheme yields lower average overhead for both the binding and the delivery operations than other methods that are described in the literature, including the current home approach, local anchoring methods, hierarchical organization of mobility agents and others. In addition, the anchor chain of a mobile host is determined dynamically with no need for preliminary definitions of static anchors or regions. This makes the proposed scheme simple and scalable.

[1] Radio resource management and or security/authentication are beyond the scope of this paper and require additional research.

The rest of the paper is organized as follows. Section 2 describes the system model. Section 3 presents the principles of the anchor chain scheme. Section 4 presents the analysis results and the upper bounds for the communication cost. Section 5 deals with some practical implementation aspects of the scheme including quality of service (QoS) support and simulation results are presented in section 6. Finally, the appendix provides a complexity analysis.

2. The model

Our model is a connectionless communication network, termed the *infrastructure graph*. This graph consists of a set of nodes V, where each node represents both a local network to which mobile hosts are attached by base stations, and a mobility agent that manages the operations of these base stations. Thus, each node $v \in V$ defines a vicinity that represents some geographical managed area. For simplicity, we assume that node v is located at the center of its vicinity, and its location is defined by the coordinates (x_v, y_v). Furthermore, the union of all the vicinities defines the *coverage area* of the system. Two nodes are called *adjacent* if their vicinities meet.

Each mobile host is associated with a *home node* and assigned a permanent address, termed the *home address*, based on its home node identifier. A mobile host may move in the coverage area. We assume a local movement assumption, i.e. a mobile host usually moves from its current vicinity only to adjacent ones. Such a transition between two vicinities is also represented as a move between the corresponding nodes. In addition, the mobile hosts also have a speed limitation, S, and therefore the minimal time a mobile host resides in a given vicinity is its diameter divided by S. Our basic distance unit is the minimal distance between two adjacent nodes. Hence, the distance between any pair of nodes is at least 1 and at most D.

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 includes an efficient routing mechanism and a reliable delivery mechanism for control messages. As a result, a control message that is sent from a source node to a destination node travels through the shortest path between the nodes. Moreover, for simplifying our calculation we assume that the propagation time of the packets is neglected with respect to the movement rate of the mobile hosts. 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 include both metrics. For any nodes u and v, we denote by $com(u, v)$ the cost of the shortest communication path and by $dist(u, v)$ the geographic distance between these nodes. In this work we assume that there is a relation between these two metrics.[2] We assume the existence of a constant $C_c > 0$, such that for each edge $(u, v) \in E$, $com(u, v) \leqslant C_c\, dist(u, v)$. C_c is termed the *correlation* constant.

3. The anchor chain scheme

3.1. The anchor chain

The proposed strategy is based on pointer forwarding and caching methods. The location of each mobile host is defined by a chain of anchors. Each anchor is a node that has been visited by the mobile host and the first anchor is the MH's home node. Every anchor records the location of the MH to a certain degree of accuracy and points to its successive node in the chain. As we follow this chain the degree of accuracy is increased, until we reach the node to which the MH is attached. Packets that are designated to the MH are sent to one of the anchors in the chain, termed the *access point*. From that node they are forwarded along the chain from one anchor to another[3] until they reach the MH attachment point, that delivers the packets to the mobile host.

A node is termed a *valid anchor* during the time that it is included in the anchor chain of the MH. The anchors are numbered in increasing order from 1 to m, where the home agent is a_1 and the current location of the MH is a_m. Let $d_k = dist(a_k, a_{k+1})$ be the geographic distance between the anchors a_k and a_{k+1}, termed the *length of the kth pointer*. The anchor chain of every MH satisfies a *length invariant*, such that for each $k > 1$, $d_k \leqslant d_{k-1}/\beta$, where $\beta > 2$ is termed the *scheme constant*. The length of d_k is upper bounded by $r_k = d_{k-1}/\beta$. Thus, r_k defines a circular region around a_k in which anchor a_{k+1} is located. This region is termed the *pointer domain* of anchor a_k. Note that the MH may be located outside the pointer domain of a valid anchor, however its distance from the anchor is bounded by $\beta/(\beta-1)r_k = d_{k-1}/(\beta-1)$ (proved by lemma 2 in the appendix). The MH keeps a record of its anchor chain and the coordinates of its anchors.[4] Hence, it can easily calculate the length of each pointer in the chain. An example of an anchor chain with $\beta = 3$ is depicted in figure 1.

3.2. The binding operation

When a MH is activated, it informs its home node of its current location. The home node sets a pointer to this location. Thus, an anchor chain with two nodes is defined for this MH. We turn to describe the algorithm for modifying an anchor chain after a movement of a MH from node u to node v. Let $\mathcal{A} = \{a_1, \ldots, a_m = u\}$ be its anchor chain just before this movement, and suppose that $\mathcal{A}$ satisfies the length invariant. During this hand-off operation, the MH obtains the address and the coordinates of its new attachment point, and informs the address to the old one. Now, the anchor chain contains $m + 1$ nodes, anchor a_{m+1} is node v, and $d_m = dist(u, v)$. However, the last pointer may violate the length invariant.

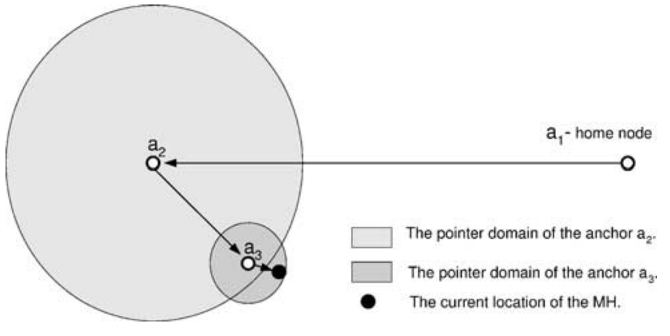

Figure 1. An example of a MH's anchor chain with $\beta = 3$.

New_Anchor_Chain procedure$(v, m, \{a_1, \ldots, a_m\})$
  $k \leftarrow m$
  modified_flag $\leftarrow$ *FALSE*
  While $((k > 1)$ and $dist(v, a_k) > dist(a_k, a_{k-1})/\beta)$ do
    Remove a_k from the anchor chain
    modified_flag $\leftarrow$ *TRUE*
    $k \leftarrow k - 1$
  End
  Return *modified_flag* and $\{a_1, \ldots, a_k, v\}$
End

Figure 2. The procedure for calculating the new anchor chain.

The MH verifies if the current anchor chain satisfies the length invariant and calculates a new chain if needed, by invoking the *New_Anchor_Chain* procedure. The procedure checks if $d_m \leqslant d_{m-1}/\beta$ for $m > 1$. Since only the last pointer was added to the chain, if this condition is fulfilled then the entire anchor chain satisfies the length invariant. Otherwise, the procedure removes in reverse order the anchors with pointers violating the length invariant. Starting with $k = m$ and with decreasing k, anchor a_k is removed from the chain, until $k = 1$ or an anchor a_k such that $dist(v, a_k) \leqslant d_{k-1}/\beta$ is found. When the loop terminates, anchor a_k is either the MH's home node or the anchor with the maximal index that node v is included in its pointer domain. Anchor a_k is termed the *modified anchor*, and it precedes node v in the received chain. A formal description of this procedure is given in figure 2.

If the *New_Anchor_Chain* procedure modified the anchor chain, the MH sends a bind message to the modified anchor for informing it about the MH's current location. This anchor sends a release message to its successive one and sets its pointer to node v. Each node that receives a release message forwards the message to its successive anchor and releases its pointer. When node u receives the release message, it sends

[2] This assumption (or a similar one) is the foundation of all the hierarchical schemes, and it becomes more realistic as new links are added to the Internet.
[3] Using tunneling techniques as described in [10].
[4] Each node is required to know only its own coordinates. When the MH visits a new anchor this information is for determining the anchor location. Thus, the NM's are not required to include GPS (or similar) devices.

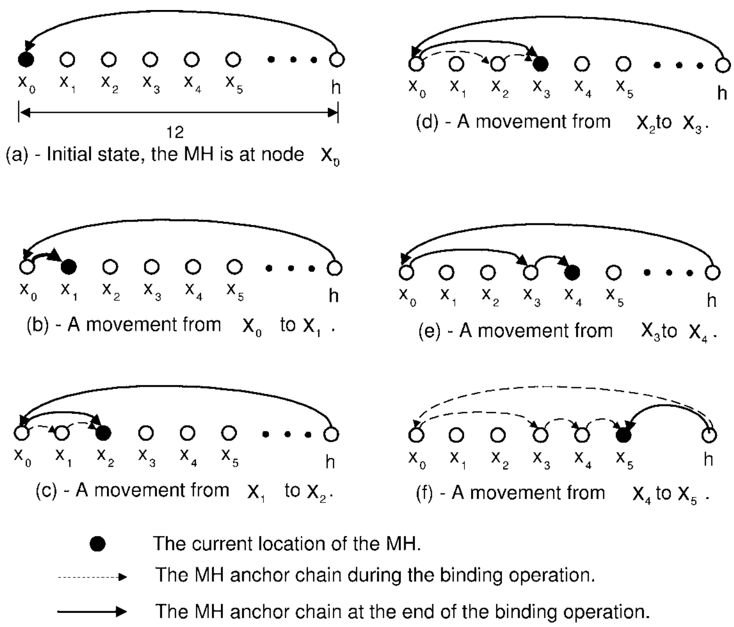

Figure 3. An example of a sequence of binding operations.

a completion message to node v, for informing it about the termination of the operation.[5]

Figure 3 presents an example of a sequence of binding operations. In this example the distance between two adjacent nodes is 1, $\beta = 3$, and the MH's home is node h. At the initial state, the MH is located at node x_0, where the distance between nodes h and x_0 is 12, and its anchor chain contains only these nodes, $a_1 = h$ and $a_2 = x_0$ (see figure 3(a)). First, the MH moves to node x_1. A new anchor, $a_3 = x_1$ is added to the chain, and the length of the second pointer is $d_2 = 1$. Since $d_2 < d_1/\beta$ $(1 \leqslant 12/3)$, the new chain remains legal, as figure 3(b) shows. Then, the MH moves to node x_2 and a pointer $d_3 = 1$ is temporarily added to the chain, as described by the dashed lines in figure 3(c). Since $d_3 > d_2/\beta$ $(1 > 1/3)$, this pointer violates the length invariant. Node x_1 is removed from the chain and node x_0 is set to point to node x_2, as drawn by the solid line in figure 3(c). A similar scenario happens when the MH moves to node x_3 (see figure 3(d)). When the MH moves to node x_4, a pointer with length $d_3 = 1$ from node x_3 to node x_4 is added to the chain. Here, the accepted chain satisfied the length invariant; $d_3 \leqslant d_2/\beta$ $(1 \leqslant 3/3)$ and $d_2 \leqslant d_1/\beta$ $(3 \leqslant 10/3)$, as depicted in figure 3(e). Finally, the MH moves to x_5 and the pointer (x_4, x_5) with length $d_4 = 1$ is added to the chain. The new chain contains 5 anchors, $\{h, x_0, x_3, x_4, x_5\}$, and it violates the length invariant (see figure 3(f)). In this case, there is no anchor a_k, $k > 1$, such that $dist(x_5, a_k) \leqslant d_{k-1}/\beta$. Therefore, the home agent, h, is set to point directly to x_5.

So far we assumed that the execution time of a binding operation is neglected with respect to the movement rate of the MH. In practice, this assumption does not always hold, and a delayed release message that reaches a valid anchor may damage the MH's anchor chain. A simple solution is associating a sequence number to each movement of the MH. This number is increased at each hand-off, and it is added as a parameter to the operation messages. When an anchor updates a pointer, it also keeps the sequence number of the associated movement, and it reacts to bind or release messages only if they carry a higher sequence number.

[5] Note that during the entire time of the binding operation a connected anchor chain is maintained. This guarantees a smooth handoff operation [11] without losses of packets designated to the MH.

3.3. The delivery operation

The delivery mechanism is composed of two operations: an *update operation* in which a MH informs its correspondent nodes (CNs) about its current anchor chain, and a *delivery operation* for transferring messages to their designated MHs. Our goal is to maintain a delivery mechanism with low communication overhead. On one hand, avoiding frequent location updates of the CNs, and on the other, delivering packets to the MH over a shortest possible path.

An efficient update strategy is obtained by reducing the overhead of the update messages. This requires synchronization between the MH and its CNs. Such synchronization is achieved by using the sequence number of the hand-off operations, as described in section 3.2. When a MH sends the first message to a CN, it also includes its current anchor chain, the anchor coordinates and the sequence number of its last hand-off operation, termed the *chain key*. The CN records this chain and the associated key. When the CN sends a packet to the MH it adds its record key. The MH attaches to the reply message the list of anchors with sequence numbers higher than the received key, and its current sequence number. This enables the CN to update its record with low overhead.

Now, consider that a CN wishes to send a message to a MH. It selects an access point from the record that it holds and sends the message to it. Then the message follows the MH's anchor chain until it reaches the MH. Note that it is always possible to send the message to the MH's home and from there to track the entire chain.[6] However, the cost may be high in comparison with the communication cost of the shortest path between the CN and the MH. Another possibility is to find a close and valid anchor based on the anchor chain record that the CN holds. From the chain definition derives that an anchor is valid if the MH is included in its pointer domain or in the pointer domain of one of its succeeding anchors. This property enables a CN to calculate a minimal period that an anchor will remain valid according to its record. Let $\{a_1, \ldots, a_m\}$ be the MH's anchor chain at a given time, where a_m is the MH location at that time. Suppose that the MH is included in the pointer domain of anchor a_k with radius $r_k = \mathit{dist}(a_{k-1}, a_k)/\beta$. Hence, its distance from the domain borders is at least $[r_k - \mathit{dist}(a_m, a_k) - D]^+$, where D is the maximal distance between two adjacent nodes and the notation $[x]^+$ means $[x]^+ = \max\{x, 0\}$. The MH will remain in this pointer domain at least for a period of $[(r_k - \mathit{dist}(a_m, a_k) - D)/S]^+$. As a result, anchor a_k will remain valid for a period τ_k, where

$$\tau_k \geqslant \left[\max_{j=1,\ldots,m} \frac{r_j - \mathit{dist}(a_j, a_m) - D}{S}\right]^+. \qquad (1)$$

Although this property provides a lower bound of the period that an anchor is valid, it cannot guarantee a low delivery cost, since for every $k > 1$ τ_k may be zero. Such an example is described in figure 3(f). There the entire anchor chain of a MH, except the home node, is changed due to a single movement.

Our delivery mechanism also uses the following properties for guaranteeing low delivery cost. Consider a movement of the MH to node v. Let $\{a_1, \ldots, a_m\}$ be the MH's anchor chain just before the movement, where d_j is the length of its jth pointer, and let a_k be the modified anchor. Then, $\mathit{dist}(a_k, v) \geqslant [(\beta - 2)/(\beta - 1)d_k - D]^+$. This property is proved by lemma 2 in the appendix. Hence, the radius, r_v, of the pointer domain of node v is at least

$$r_v = \frac{\mathit{dist}(a_k, v)}{\beta} \geqslant \left[\frac{1}{\beta}\left(\frac{\beta - 2}{\beta - 1} d_k - D\right)\right]^+$$

and the duration that node v is a valid anchor, denoted by τ_v, is at least

$$\tau_v \geqslant \frac{r_v}{S} \geqslant \left[\frac{1}{\beta S}\left(\frac{\beta - 2}{\beta - 1} d_k - D\right)\right]^+.$$

Our scheme uses these properties for defining valid access points that are not necessarily valid anchors. A node is termed a *valid access point* if it is a valid anchor or it points to a valid anchor. When an anchor a_j, $j > k$, receives a release message it sets a temporary pointer to node v for a period of

$$\tilde{\tau}_k = \left[\frac{1}{\beta S}\left(\frac{\beta - 2}{\beta - 1} d_{k-1} - D\right)\right]^+. \qquad (2)$$

Since, for each $j > k$, $d_{j-1} \leqslant d_k$, node a_j remains a valid access point during that time.

[6] The home can also be used if it turns out that the access point is not valid from any reason.

```
Select_Access_Point procedure(Δt, m, {a_1, ..., a_m})
   k ← m
   τ ← [(d_{m-1}/β − D)/S]^+
   While (k > 1) and
      (Δt > τ + [1/(βS)((β − 2)/(β − 1)d_{k-1} − D)]^+) do
      k ← k − 1
      τ ← max{τ, [(d_{k-1}/β − dist(a_m, a_k) − D)/S]}
   End
   Return a_k
End
```

Figure 4. The procedure for selecting the access point.

Now, suppose that a CN wishes to send a message to the MH. Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be the record of the MH's anchor chain that it holds, and let Δt[7] be the time that elapsed since it cached $\mathcal{A}$. The selected access point is anchor $a_k \in \mathcal{A}$ with the highest index k such that $\Delta t \leqslant \tau_k + \tilde{\tau}_k$. Where, τ_k lower bounds the time that anchor a_k will remain a valid anchor as it is calculated by equation (1) and $\tilde{\tau}_k$ is the period that it will hold a temporary pointer to a valid anchor after been removed from the chain, according to equation (2). This requirement guarantees that the message is sent to a valid access point that is close to the current location of the MH. A formal description of the access point selection algorithm is given in figure 4 for a chain with two or more anchors.

In practice, our scheme can be improved by using a probabilistic approach. Each time a CN sends a message to a MH, it calculates two access points. A deterministic access point that is selected according to the above method, and a probabilistic access point that is determined according to some probability heuristic. For instance, estimating the nearest access point according to the MH's mean velocity. The CN sends the message to the probabilistic access point, and if this node is not a valid access point, then the message is forwarded to the determined access point.

4. The results of the complexity analysis

This section presents the results of the worst case analysis for the binding and delivery operations. The complexity analysis can be found in the appendix. We use the following characteristics for the system analysis. Let the infrastructure graph be a graph with C_c correlation constant. Thus, the communication cost of sending a message from node v to u is at most $C_c\,\mathit{dist}(u, v)$, as described in section 2. In addition, let C_a be the cost of accessing a mobility agent. We also denote by β our scheme constant, where $\beta > 2$.

Consider a MH, we denote by $\mathit{dist}(h, MH)$ the geographic distance between the MH home node, h, and its current location. The communication cost of following the MH anchor chain, starting from its home node until reaching its current location is $O(C_c(\beta/(\beta - 2))\,\mathit{dist}(h, MH))$ and the number of anchors in the chain is $O(\log_\beta \mathit{dist}(h, MH))$.

[7] Δt should also include the round trip delay.

Let *Total_Move* be the total distance that the MH traveled. Hence, the total cost of all the binding operations is

$$O\bigg(C_c\beta\, Total\_Move \log_\beta Total\_Move + C_a \frac{\beta}{(\beta-1)^2} Total\_Move\bigg).$$

We turn to calculate the overhead of delivering a message to a MH. Suppose that a CN holds a record of the MH anchor chain from time t_0, and it sends a message to this MH after a period of Δt. Let $Com(CN, MH)'$ be the communication cost of sending a message from the CN to the MH over the shortest path at time $t_0 + \Delta t$. The total cost of the delivery operation is

$$O\big(Com(CN, MH)' + C_c S\Delta t + C_a \log(S\Delta t)\big).$$

This shows that the additional overhead to $Com(CN, MH)'$ is relative to $S\Delta t$, and when Δt is small the delivery overhead is close to optimal.

5. Implementation aspects

This section presents some practical improvements to the proposed scheme for further reducing the scheme overhead, increasing its reliability.

5.1. Actual and virtual anchors

In the proposed scheme each valid anchor has both a geographic and communication roles. Its geographic location defines the center of its pointer domain, and it also points to the location of its successive anchor. These two roles are independent and can be implemented separately. Consider a star network that is connected to a general network, as figure 5 describes. Suppose that one of the MH anchors is located at node v, and its pointer domain is the circular region around it, denoted with a gray background. Every message designated to node v is routed via node u. Therefore, it is recommended to hold the pointer to the successive anchor at node u instead of node v for reducing the communication cost of both the delivery and the binding operations. Node v is termed a *virtual anchor*, and it geographic location determines the pointer domain, while node u, termed the *actual anchor*, is the actual node that is included in the anchor chain. The anchor chain of a MH is defined by a set of actual anchors that hold the chain pointers and the geographic location of their corresponding virtual anchors that define the chain pointer domains. This partition reduces the overhead of the binding and delivery operations. It also increases the network reliability, since the actual anchors are internal routers that are selected according to some robustness criteria.

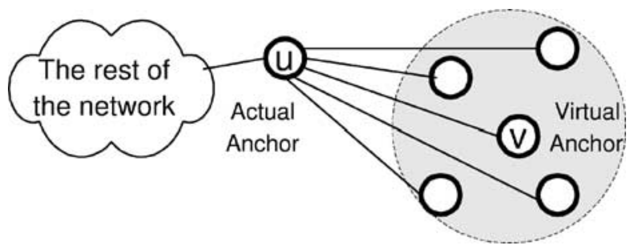

Figure 5. A separation to virtual and actual anchor.

5.2. Response to failures

Using the actual anchor partition method increases the system reliability, however the scheme is still sensitive to anchor failures. An anchor failure can be easily detected by periodically sending query and reply messages between the MH and its home node along the anchor chain. However, in case of failure this method cannot detect the malfunctioning anchor. Our approach is based on the following observation. Let $\mathcal{A}$ be the anchor chain when a node v becomes a valid anchor, during the time period that node v is a valid anchor all the nodes in $\mathcal{A}$ are also valid anchors. In our method when a node becomes a valid anchor it is informed about its two predecessor anchors in the chain, that are termed the *parent* and the *grandparent* anchors. Each valid anchor periodically checks whether its parent anchor is functioning. When it detects a failure, it informs its grandparent anchor about the failure and the parent anchor is removed from the chain. Finally, a proper update message is send along the chain for updating its successor anchor and the MH. This method uses local detection are repairing mechanism that guarantees that the chain remains connected.

5.3. Quality of Service support

Quality of Service (QoS) becomes a genuine need in mobile-IP networks, since many new applications require guaranteed QoS parameters like delay and bandwidth [4]. In practice, mobility and QoS support are contradicting requirements. The current methods for providing QoS support (like MPLS and Diffserv) are based of finding and maintaining fixed paths that satisfies the QoS constraints between the connection end-users. This approach contradicts the nomadic behavior of the mobile hosts, where the paths between the users are constantly changed. Still, our scheme can support QoS requirement better than other proposed scheme from the following reasons. It is much easier to find paths that satisfy the QoS constraints when the connection end-points are close to each other. In our scheme the route from a corresponding node (CN) to a mobile host contains two parts. The first is the path between the CN and a valid access point of the MH, while the rest of the route is a small tail of the MH's anchor chain. Generally, the longer component of this route is the first path, which sustains for a relative long period of time. The second part, the chain tail, is frequently modified, but it is composed of short segments that can be easily replaced.

6. Numerical results

We compared by simulations the performance of the anchor chain scheme with four other methods over different net-

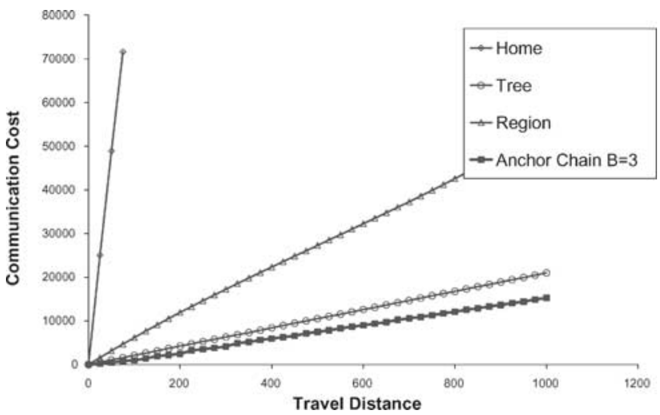

Figure 6. The total communication cost of a binding operation sequence.

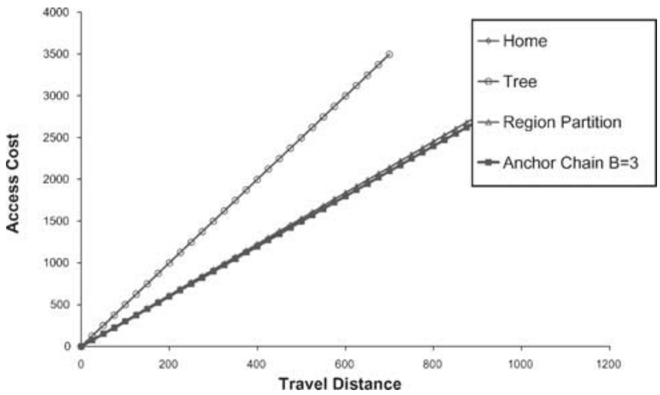

Figure 7. The total access cost of a binding operation sequence.

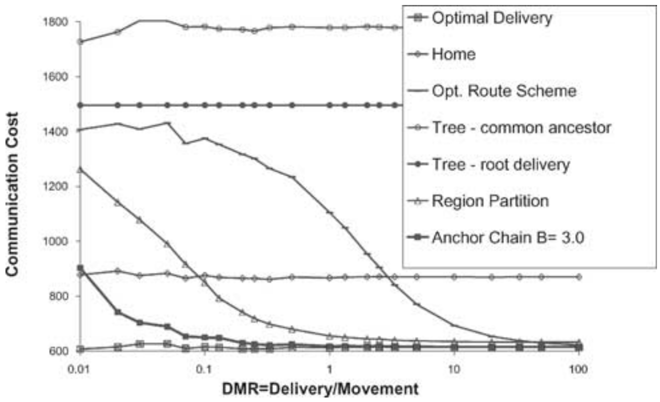

Figure 8. The average communication cost of a delivery operation.

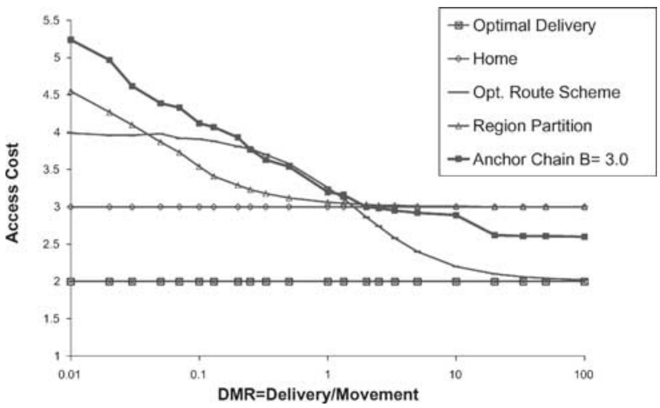

Figure 9. The average access cost of a delivery operation.

works. The evaluated schemes were:

- The *home* approach [9], where the home agent is located at the geographic center of the graph.
- The *optimal route* scheme [13] with the same binding operation as the home method. Each CN caches the location of its peer MHs, and sends messages directly to these nodes. If a peer MH moved to another node the message is forwarded to the MH home agent.
- The *region partition* approach [6,7], where the coverage area of the network is divided into square regions.[8] Each region includes an agent that tracks the location of the MHs in its area and the home agent of each MH points to the region where it is located. We use a delivery mechanism similar to [7]. The CN holds records of its peer MH regions and delivers messages to these region agents. If a peer MH left the region, the message is forwarded to the MH home agent, which delivers it to the current region agent of the MH.
- The *tree* method [12,16] where the tree root is located at the geographic center of the graph, each interior control node is associated with a square region and it has four children. We simulated two tree delivery methods. The first is based on finding the lowest common ancestor for the CN and the MH as described in [16]. In the second, the messages are sent to the tree root and from there they track the MH pointers until they reach the MH attachment points, similar to [12].
- The *anchor chain* scheme as described in section 3. Initially, we present our simulation results for $\beta = 3$ that achieves low overhead for both the binding and the delivery operations. Then we evaluate the effect of β over the scheme performance.

We evaluated the total cost of sequences of binding operations as a result of roaming different distances, and the average cost of delivery operations with various *delivery–movement ratio* ($DMR = delivery/movement$), also termed *call–movement ratio* (CMR) [7]. Selected typical results from our experiments are depicted in figures 6–9. The tested communication network is a grid graph with 2^{20} nodes (1024×1024), where both the communication cost and the distance between two adjacent nodes is 1. Figures 6 and 7 depict the results for the binding operations. They show that for any travel distance the anchor chain method yields the lowest communication overhead and with minimal number of accesses to mobility agents. Figures 8 and 9 show the average cost of a delivery operation of the above methods[9] for different DMR values. The MH travels a random path with half

[8] We evaluate the overheads yield by partitioning the coverage area into various square sizes.

[9] The delivery cost of the tree method is omitted due to its high overhead.

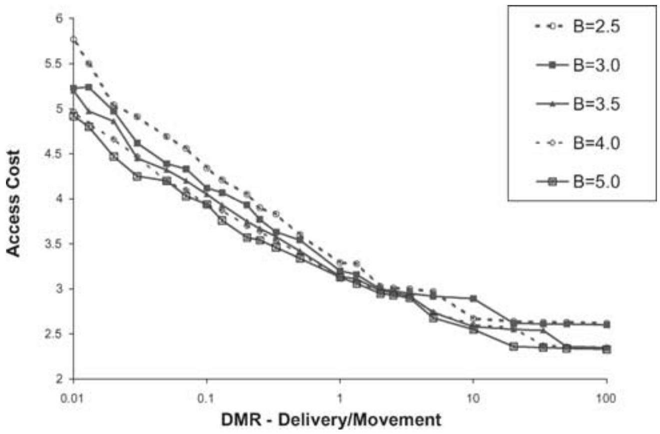

Figure 10. The average access cost of an anchor chain delivery operation for different value of β.

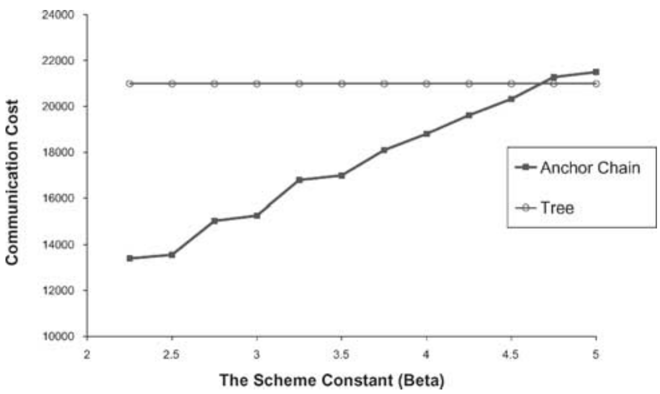

Figure 11. The average communication cost of the anchor chain scheme with various values of β after traveling 1000 units.

S movement velocity and the CNs are uniformly distributed over the graph nodes. We assume that the response messages inform the CNs about the MH location, and the round trip time is shorter then the period between two successive deliveries. For comparison, the diagram also presents the overhead of the *optimal delivery*, in which a CN sends its messages directly to the location of its peer MH. These figures shows that the anchor chain method produces the minimal communication overhead in compare to the other schemes, and with limited number of accesses to mobility agents. Although, figure 9 shows that for low DMR the access cost is relatively high, in practice this is not the case and the delivery rate is much higher then the movement rate. Moreover, for high DMR the delivery overhead of the anchor chain scheme is almost optimal. These simulations indicate that the proposed scheme achieves low average overhead for both the delivery and binding operations.

We also checked the effect of β over the scheme performance. We simulated both the delivery and the binding operation with various values of β in the range between 2.25 to 5. In the case of delivery operation we notice that the value of β has only a minor empact over the system performance, generally the access cost is decreased by increasing β, as it is depicted in figure 10. In the case of binding operations we observed that the average communication cost of a sequence of binding operations is increased by increasing β, as it is shown in figure 11. The access cost is almost unchanged.

Appendix. Complexity analysis

This section presents the worst case analysis for the binding and delivery operations. Due to space limitation some proofs are omitted and can be found in [3]. We use the following characteristics for the system analysis. Let the infrastructure graph be a graph with C_c correlation constant. Thus, the communication cost of sending a message from node v to u is at most $C_c\, dist(u, v)$, as described in section 2. In addition, let C_a be the cost of accessing a mobility agent. We assume that the distance between two adjacent nodes is at least 1, and no more them D. We also denote by β our scheme constant, where $\beta > 2$.

Consider a MH and let $\mathcal{A} = \{a_1, \dots, a_m\}$ be its anchor chain at a given time. Node a_1 is its home node h, a_m is its location at the given time, and $d_k = dist(a_k, a_{k+1})$ is the length of the kth pointer. According to the length invariant, d_k is upper bounded by $r_k = d_{k-1}/\beta$. r_k is termed the *radius* of anchor a_k, and it defines a circular region around anchor a_k that contains anchor a_{k+1} and is termed the *pointer domain* of anchor a_k. Let $Tail\_Length_k = \sum_{j=k}^{m} d_j$ be the length of the chain tail starting at anchor a_k, and let $Chain\_Length = Tail\_Length_1$ be the total length of the chain. We denote by $dist(v, MH)$ the geographic distance between node v the MH location. These parameters have the following properties.

Lemma 1 [3]. For each valid anchor a_k, $1 \leqslant k < m$,

$$Tail\_Length_k \leqslant \frac{\beta}{\beta - 1} d_k.$$

Corollary 1. For each valid anchor a_k, $1 < k < m$,

$$Tail\_Length_k \leqslant \frac{d_{k-1}}{\beta - 1}.$$

Lemma 2 [3]. For each valid anchor a_k, $1 \leqslant k < m$,

$$\frac{\beta - 2}{\beta - 1} d_k \leqslant dist(a_k, MH) \leqslant \frac{\beta}{\beta - 1} d_k.$$

Theorem 1. Consider a MH with home node h. Then,

$$Chain\_Length \leqslant \frac{\beta}{\beta - 2}\, dist(h, MH)$$

and the number of anchors in its chain is at most

$$2 + \log_\beta(\beta - 2) + \log_\beta dist(h, MH).$$

Proof. From lemma 1, $Chain\_Length \leqslant (\beta/(\beta - 1))d_1$. According to lemma 2, $d_1 \leqslant (\beta - 1)/(\beta - 2)\, dist(h, MH)$.

Hence, $Chain\_Length \leqslant \beta/(\beta-2)\, dist(h, MH)$. According to the length invariant, the number of anchors is bounded by

$$1+\log_\beta\left\{\frac{\beta}{\beta-2}\, dist(h, MH)\right\} = 2+\log_\beta(\beta-2)+\log_\beta dist(h, MH). \qquad \Box$$

As a result of theorem 1, the communication cost of following a MH anchor chain, starting from its home node until reaching its current location is $O(C_c\beta/(\beta-2)\, dist(h, MH))$ and the number of anchors in the chain is $O(\log_\beta dist(h, MH))$.

We turn to calculate the worst case amortized cost of the binding operations. When a MH is activated, its home node sets a pointer to its current location. Since this operation is performed only once, we ignore its communication cost. After its activation the MH is free to move from place to place. In its way the MH triggers a sequence of binding operations. First, we calculate the cost of a single binding operation in which the MH moves from node u to node v, where $\mathcal{A} = \{a_1, \ldots, a_m = u\}$ is the MH's anchor chain just before the movement. Let $a_{k^*} \in \mathcal{A}$ be the anchor with the minimal index that node v is not included in its pointer domain. It is termed the *released chain head*, and r_{k^*} denotes the radius of its pointer domain. Note that a_{k^*-1} is the modified anchor and the anchors a_k with $k \geqslant k^*$ are released from the chain. In addition, we denote by *Move* the distance that the MH has traveled during the time period that a_{k^*} was a valid anchor.

Lemma 3 [3]. *Move* $> r_{k^*}$ and *Move* $\geqslant dist(u, v) + Tail\_Length_{k^*}$.

Lemma 4. The communication cost of the binding operation is upper bounded by

$$2C_c(\beta+1)Move.$$

Proof. During the binding operation, node u sends a bind message to the modified anchor, a_{k^*-1}. Anchor a_{k^*-1} sends a release message, that follows the anchor chain until it reaches node v, Then a completion message is sent to node u. According to the triangle inequality, lemma 1 and lemma 3, the distance that the bind message traveled is:

$$\begin{aligned} dist(v, a_{k^*-1}) &\leqslant dist(u, v) + dist(u, a_{k^*}) \\ &\leqslant dist(u, v) + Tail\_Length_{k^*} + d_{k^*-1} \\ &\leqslant Move + d_{k^*-1} \\ &\leqslant Move + \beta r_{k^*} \leqslant (\beta+1)Move. \end{aligned}$$

In a similar way, the distance that the release and completion messages traveled is:

$$Tail\_Length_{k^*-1} + dist(u, v) \leqslant (\beta+1)Move.$$

Hence, the communication cost of this operation is bounded by $2C_c(\beta+1)Move$. $\Box$

Now, let us calculate the number of accesses to mobility agents during the operation.

Lemma 5. The number of nodes that were accessed due to the binding operation is at most

$$3 + \lfloor \log_\beta r_{k^*} \rfloor.$$

Proof. The nodes that were accessed in this operation are node v, the modified anchor a_{k^*-1} and all the nodes in the released chain. The number of nodes in this chain is at most $1 + \lfloor \log_\beta r_{k^*} \rfloor$. Thus, the total number of accessed nodes is at most $3 + \lfloor \log_\beta r_{k^*} \rfloor$. $\Box$

We turn to calculate the overhead of all the binding operations. For that purpose we associate with each binding operation a sequence number n, $1 \leqslant n \leqslant n_{\max}$, and we denote by t_n the time when the nth operation occurred. Anchor a_n represents the released chain head at the nth operation, and let τ_n and r_n be the time when a_n became a valid anchor and its radius, respectively. Let $Move(\tau, t)$ be the distance that the MH traveled during that period $[\tau, t]$. *Total_Move* is the total distance the MH traveled, and let $\delta = \lfloor \log_\beta Total\_Move \rfloor$.

We denote by Λ_p, $p \in \mathcal{Z}^+$, the set of sequence numbers n such that $\beta^p \leqslant r_n < \beta^{p+1}$, thus

$$\Lambda_p = \{n \mid \beta^p \leqslant r_n < \beta^{p+1}\}.$$

This set represents all the binding operations in which the radius r_n of the released chain head is in the range $[\beta^p, \beta^{p+1})$.

Lemma 6. For every $p, q \in \mathcal{Z}^+$, $p \neq q$, holds:

$$\Lambda_p \cap \Lambda_q = \emptyset \quad \text{and} \quad \bigcup_{p=0}^{\delta} \Lambda_p = \{1, \ldots, n_{\max}\}.$$

Proof. Each $n \in \{1, \ldots, n_{\max}\}$ defines a single value $r_n \leqslant$ *Total_Move*, and it is included only in a single set Λ_p. The complete proof can be found in [3]. $\Box$

Lemma 7 [3]. For every $p \in \mathcal{Z}^+$ and every $n, k \in \Lambda_p$,

$$[\tau_n, t_n) \cap [\tau_k, t_k) = \emptyset.$$

Corollary 2. For every $p \in \mathcal{Z}^+$,

$$\sum_{n\in\Lambda_p} Move(\tau_n, t_n) \leqslant Total\_Move.$$

Lemma 8. For every $p \in \mathcal{Z}^+$,

$$|\Lambda_p| \leqslant \left\lfloor \frac{Total\_Move}{\beta^p} \right\rfloor.$$

Proof. From lemma 3, for each $n \in \Lambda_p$, $Move(\tau_n, t_n) > r_n \geqslant \beta^p$. From corollary 2, $\sum_{n\in\Lambda_p} Move(\tau_n, t_n) \leqslant Total\_Move$. Hence, $|\Lambda_p| \leqslant \lfloor Total\_Move/\beta^p \rfloor$. □

Corollary 3. For every $p > \delta$, $|\Lambda_p| = 0$.

Theorem 2. The total communication cost of the binding operations is upper bounded by

$$2C_c(\beta + 1)Total\_Move(1 + \log_\beta Total\_Move).$$

Proof. Let *Total_Com_Cost* be the total communication cost of all the binding operations. Let *Com_Cost*(n) be the cost of the nth operation. From lemmas 4–6 and corollaries 2, 3,

$$\begin{aligned} &Total\_Com\_Cost \\ &\quad = \sum_{p=0}^{\delta} \sum_{n\in\Lambda_p} Com\_Cost(n) \\ &\quad \leqslant 2C_c(\beta + 1) \sum_{p=0}^{\delta} \sum_{n\in\Lambda_p} Move(\tau_n, t_n) \\ &\quad \leqslant 2C_c(\beta + 1) \sum_{p=0}^{\delta} Total\_Move \\ &\quad \leqslant 2C_c(\beta + 1)Total\_Move(1 + \log_\beta Total\_Move). \end{aligned}$$ □

Theorem 3. The maximal number of accesses to mobility agents is

$$\left(3 + \frac{\beta}{(\beta - 1)^2}\right) Total\_Move.$$

Proof. Let *Total_Acc_Num* be the number of accesses to mobility agents during all the binding operations, and let *Acc_Num*(n) be the number of accesses at the nth operation. From the definition of the set Λ_p and lemmas 5–7,

$$\begin{aligned} &Total\_Acc\_Num \\ &\quad = \sum_{p=0}^{\delta} \sum_{n\in\Lambda_p} Acc\_Num(n) \\ &\quad \leqslant \sum_{p=0}^{\delta} \sum_{n\in\Lambda_p} (3 + \lfloor \log_\beta r_n \rfloor) \\ &\quad \leqslant 3n_{\max} + \sum_{p=0}^{\delta} \sum_{n\in\Lambda_p} p \leqslant 3n_{\max} + Total\_Move \sum_{p=0}^{\infty} \frac{p}{\beta^p}. \end{aligned}$$

The minimal distance between two adjacent nodes is 1. Thus, $n_{\max} \leqslant Total\_Move$. In addition, we use the equation

$$\sum_{i=0}^{\infty} \frac{i}{\beta^i} = \frac{\beta}{(\beta - 1)^2}.$$

Hence, the total number of accesses is at most

$$\left(3 + \frac{\beta}{(\beta - 1)^2}\right) Total\_Move.$$ □

As a result of theorems 2 and 3, the total cost of all the binding operations is

$$O\left(C_c\beta\, Total\_Move \log_\beta Total\_Move + C_a \frac{\beta}{(\beta - 1)^2} Total\_Move\right).$$

We turn to calculate the overhead of delivering a message to a MH. Suppose that a CN holds a record, $\mathcal{A} = \{a_1, \ldots, a_m\}$, of a MH anchor chain from time t_0, and it sends a message to this MH after a period of Δt. Note that the maximal distance that the MH has traveled during this period is at most $\Delta t S + D$. Let $a_k \in \mathcal{A}$ be the selected access point for delivering the message. The CN selects this node based on the anchor chain $\mathcal{A}$ and the elapsed time Δt. To distinguish between measurements at time t_0 and at time $t_0 + \Delta t$, we denote parameters that are measured at time $t_0 + \Delta t$ with quote sign, i.e. $dist(a, MH)$ represents the distance between the MH and node a at time t_0, and $dist(a, MH)'$ represents the distance at time $t_0 + \Delta t$. In addition, let $Com(a, b)'$ be the communication cost of sending a message from node a to node b over the shortest path at time $t_0 + \Delta t$, where one of the nodes may be the MH location at that time.

Lemma 9. Let τ_k be the minimal period time that anchor a_k will remain a valid anchor after t_0, as it is calculated by equation (1) and let $\tilde{\tau}_k$ be the period that it will hold a temporary pointer to a valid anchor after been removed from the chain, according to equation (2). If $\Delta t \leqslant \tau_k + \tilde{\tau}_k$ then node a_k is a valid access point.

Proof. When the CN recorded the anchor chain $\mathcal{A}$, at time t_0, every node $a_k \in \mathcal{A}$ was a valid anchor of the MH. From the chain definition derives that every anchor $a_k \in \mathcal{A}$ will remain valid anchor until the MH will leave the its pointer domain and the pointer domains of all its succeeding anchors. Thus a_k will remain valid anchor for at least a period of τ_k, defined by equation (1),

$$\tau_k \geqslant \left[\max_{j=1,\ldots,m} \frac{r_j - dist(a_j, a_m) - D}{S}\right]^+,$$

where D is the maximal distance between two adjacent nodes and the notation $[x]^+$ means $[x]^+ = \max\{x, 0\}$.

Consider a given node $a_k \in \mathcal{A}$. If at time $t_0 + \Delta t$ it is still a valid anchor then the lemma is satisfied. Otherwise, a_k was removed from the chain due to a movement of the MH to a node v, exterior to the pointer domain of a_k, at some time after $t_0 + \tau_k$. At that time, node a_k sets a temporary pointer to node v for a period of $\tilde{\tau}_k$ as it is defined by equation (2), where

$$\tilde{\tau}_k = \frac{1}{\beta S}\left[\frac{\beta - 2}{\beta - 1} d_{k-1} - D\right]^+.$$

Let $a_j \in \mathcal{A}$ be the modified anchor of this hand-off operation. From lemma 2 and since $j < k$ therefore

$$d_j \geqslant d_{k-1}, \qquad dist(a_j, v)' \geqslant \left[\frac{\beta-2}{\beta-1}d_{k-1} - D\right]^+.$$

Therefore, the radius of the pointer domain of node v is

$$r_v \geqslant \frac{1}{\beta}\left[\frac{\beta-2}{\beta-1}d_{k-1} - D\right]^+$$

and node v is a valid anchor at least for a period of

$$\frac{r_v}{S} \geqslant \frac{1}{\beta S}\left[\frac{\beta-2}{\beta-1}d_{k-1} - D\right]^+ \geqslant \tilde{\tau}_k.$$

As a result if $\Delta t \leqslant \tau_k + \tilde{\tau}_k$ then node a_k is either, a valid anchor or it points to a valid anchor of the MH. □

Let $path(a_k, MH)'$ be the total length of all the pointers in the path from node a_k to the location of the MH at time $t_0 + \Delta t$.

Lemma 10. $path(a_k, MH)'$ is at most $(\beta^2/(\beta-2)+1)\Delta t S + 2(\beta-1)/(\beta-2)D$.

Proof. Recall that the selected access point a_k, is the anchor in $\mathcal{A}$ with the highest index that satisfies the condition of lemma 9. Hence, $d_k \leqslant (\beta-1)/(\beta-2)[\beta\Delta t S + D]$. Thus,

$$\begin{aligned} path(a_k, MH)' &\leqslant Tail\_Length_k + \Delta t S + D \\ &\leqslant \frac{\beta}{\beta-1}d_k + \Delta t S + D \\ &\leqslant \left(\frac{\beta^2}{\beta-2} + 1\right)\Delta t S + \frac{2(\beta-1)}{\beta-2}D. \end{aligned}$$ □

Theorem 4. The total communication cost of the delivery operation is upper bounded by

$$Com(CN, MH)' + 2C_c\left\{\left(\frac{\beta^2}{\beta-2} + 1\right)\Delta t S + \frac{2(\beta-1)}{\beta-2}D\right\}.$$

Proof. The communication cost of the delivery operation is $Com(CN, a_k)' + Com(path(a_k, MH)')$. According to the triangle inequality,

$$\begin{aligned} Com(CN, a_k)' &\leqslant Com(CN,MH)' + Com(MH, a_k)' \\ &\leqslant Com(CN,MH)' + Com\big(chain\_of\_node\_a_k{}'\big). \end{aligned}$$

Using lemma 10,

$$\begin{aligned} &Delivery\_Cost' \\ &\quad\leqslant Com(CN, MH)' + 2Com\big(path(a_k, MH)'\big) \\ &\quad\leqslant Com(CN, MH)' \\ &\qquad + 2C_c\left[\left(\frac{\beta^2}{\beta-2} + 1\right)S\Delta t + \frac{2(\beta-1)}{\beta-2}D\right]. \end{aligned}$$ □

Theorem 5. The total number of accesses to mobility agents is at most

$$1 + \log_\beta\left\{\left(\frac{\beta^2}{\beta-2} + 1\right)\Delta t S + \frac{2(\beta-1)}{\beta-2}D\right\}.$$

Proof. If the access point a_k is a valid anchor, the number of accesses is at most

$$\big(1 + \log_\beta\big(Tail\_Length_k'\big)\big).$$

According to lemmas 2 and 10,

$$Tail\_Length_k' \leqslant \left(\frac{\beta^2}{\beta-2} + 1\right)\Delta t S + \frac{2(\beta-1)}{\beta-2}D.$$

As a result the total number of accesses is at most

$$1 + \log_\beta\left\{\left(\frac{\beta^2}{\beta-2} + 1\right)\Delta t S + \frac{2(\beta-1)}{\beta-2}D\right\}.$$

If node a_k points to a valid anchor v then the accessed nodes are node a_k and all the nodes at the MH anchor chain from node v. Thus, $Access\_num \leqslant 2 + \log_\beta(\Delta t S)$. □

As a result of theorems 4 and 5 the total cost of the delivery operation is

$$O\big(Com(CN, MH)' + C_c S\Delta t + C_a \log(S\Delta t)\big).$$

We proved that the additional overhead to $Com(CN, MH)'$ is relative to $S\Delta t$, and when Δt is small the delivery overhead is close to optimal.

References

[1] B. Awerbuch and D. Peleg, Concurrent online tracking of mobile users, in: *SIGCOM Symposium on Communication, Architectures and Protocols* (October 1991).

[2] Y. Bejerano and I. Cidon, An efficient mobility management strategy for personal communication systems, in: *Proceedings of MobiCom98* (1998).

[3] Y. Bejerano and I. Cidon, An anchor chain scheme for IP mobility management, Research Report (1999).

[4] H. Chaskar (ed.), Requirements of a QoS solution for mobile IP, Internet Draft, `draft-ietf-mobileip-qos-requirements-01.txt` (August 2001), work in progress.

[5] C. Cho and L.F. Marshell, An efficient location and routing scheme for mobile computing environment, IEEE Journal on Selected Areas in Communications 13(5) (1995).

[6] E. Gustafsson, A. Jonsson and C.E. Perkins, Mobile IPv4 regional registration, Internet Draft, `draft-ietf-mobileip-reg-tunnel-05.txt` (September 2001), work in progress.

[7] J.S.M. Ho and F. Akyildiz, Dynamic hierarchical database architecture for location management in PCS networks, IEEE/ACM Transactions on Networking 5(5) (1995) 187–196.

[8] R. Jain, Y.B. Lin, C. Lo and S. Mohan, A forwarding strategy to reduce network impacts of PCS, in: *Proceedings of IEEE INFOCOM* (1995).

[9] C.E. Perkins (ed.), IP mobility support, RFC 2002 (October 1996).

[10] C.E. Perkins, IP encapsulation within IP, RFC 2003 (October 1996).

[11] C.E. Perkins, Mobile IP, IEEE Communication Magazine 35(5) (1997).

[12] C.E. Perkins, Mobile-IP local registration with hierarchical foreign agents, Internet Draft, `draft-perkins-mobileip-hierfa-00.txt` (February 1996), work in progress.

[13] C.E. Perkins and D.B. Johnson, Route optimization in mobile IP, IEEE Communication Magazine 34(9) (1996), also available as Internet Draft, draft-ietf-mobileip-optim-07.txt (November 1997), work in progress.

[14] H. Soliman, C. Castelluccia, K. El-Malki and L. Bellier, Hierarchical MIPv6 mobility management (HMIPv6), Internet Draft, draft-ietf-mobileip-hmipv6-04.txt (June 2001), work in progress.

[15] F. Teraoka, Y. Yokote and M. Tokoro, A network architecture providing host migration transparency, in: *Proceedings of ACM SIGCOMM* (September 1991) pp. 209–220.

[16] J.Z. Wang, A fully distributed location registration strategy for universal personal communication systems, IEEE Journal on Selected Areas in Communications 11(6) (1993).

Yigal Bejerano received his B.Sc. in computer engineering in 1991 (sumnia cum laude), his M.Sc. in computer science in 1995, and his Ph.D. in electrical engineering in 2000, from the Technion – Israel Institute of Technology, Haifa, Israel. He is currently a member of the technical staff (MTS) at Bell Laboratories, Lucent Technologies. His research interests are mainly management aspects of high-speed and wireless networks, including the areas of mobility management, network monitoring, topology discovery and QoS routing. Dr. Bejerano is on the technical program committee (TPC) of the INFOCOM-2002 and INFOCOM-2003 conferences.
E-mail: bej@research.bell-labs.com

Israel Cidon is a Professor in the Faculty of Electrical Engineering at the Technion. He holds a D.Sc. and a B.Sc. in electrical engineering from the Technion (1980 and 1984, respectively). His research interests are in the field of high-speed networks, network architecture, network control, quality of service and distributed algorithms. In 1994 and 1995, he was manager of High-Speed Networking at Sun Microsystems Labs, CA. There, he founded and lead Sun's first high-speed networking research group. Between 1985 and 1994, he was with by IBM Thomas J. Watson Research Center, NY where he served as a Research Staff Member and a Manager of the Network Architectures and Algorithms Group. He was responsible for projects in high-speed networks, switch design, network control and wireless networks. In particular he lead the implementation of PARIS the world first integrated packetized network to carry data, voice and video and MetaRing the first spatial reuse optical ring. He was also instrumental in the technology development of Micronet Ltd., a vendor of data entry terminals, Viola Networks, a provider of distributed network performance verification and fault isolation software and VersEdge Technologies, a provider of a software solution that accelerates file sharing over enterprise Intranets/Extranets. Israel Cidon was a founding editor for the IEEE/ACM Transactions on Networking, Editor for Network Algorithms for the IEEE Transactions on Communications and a guest editor for Algorithmica. He is a recipient of the IBM Outstanding Innovation Award for his work on the PARIS project and topology update algorithms (1989 and 1993, respectively). He has authored over 100 papers in leading journals and conferences and 15 US patents.
E-mail: cidon@ee.technion.ac.il