# Performance Evaluation of a Multilevel Hierarchical Location Management Protocol for Ad Hoc Networks

Sumesh J. Philip [*], Joy Ghosh, Chunming Qiao

*Department of Computer Science and Engineering, State University of New York at Buffalo, Amherst NY 14260*

---

## Abstract

In networks where nodes are aware of their locations via the use of a GPS receiver or other localization techniques, geographic routing has been suggested to be the candidate of choice for robust and resource efficient routing. However, efficient location management algorithms are required for the source node to obtain the destination node's location before it can start data transfer using geographic routing. To be deemed scalable with respect to network size, mobility and traffic, the signalling overhead due to location management must be kept low so that the performance of geographic routing is minimally affected. This paper describes the performance of a novel multi-level hierarchical grid location management protocol that we call HGRID, for large scale ad hoc networks. We show that the average per node signalling cost in HGRID grows only logarithmically in the total number of nodes in a uniformly randomly distributed network - a substantial improvement over the signalling cost incurred by current location management schemes. We also carry out extensive simulations to quantitatively compare the performance of the protocol against other well known location management protocols, and to study how location management can affect the scalability of geographic routing. Results show that our protocol outperforms others in terms of network throughput and end–to–end packet delay with increasing network size or average node speed. Thus hierarchical grid location management scales well for large scale mobile ad hoc networks.

*Key words:* Ad hoc networks, Geographic routing, Location/Mobility management, Probabilistic analysis, Performance analysis

---

[*] Correspondence Author:
*Email address:* `sumeshjp@cse.buffalo.edu` (Sumesh J. Philip ).

*Preprint submitted to Elsevier Science* *14 June 2004*

# 1 Introduction

Routing in ad hoc networks has posed an interesting challenge in the research community, in which finding and maintaining a route between a source–destination pair in a communication session imposes a major hurdle in designing an efficient routing protocol for such networks. Due to mobility, the network topology varies frequently, and end-to-end sessions are subject to link failures constantly. While many solutions have been proposed for routing in mobile ad hoc networks [1–6], few have considered the issue of scalability of such protocols in networks having node membership in the order of thousands, and spread over a large geographic area. A unique characteristic of ad hoc networks is that the limited bandwidth of the wireless channel is shared by signalling traffic as well as data, and the former is given a higher priority than data. This works fairly well for current routing protocols in small networks with low node mobility, since the volume of signalling traffic is low enough to carry out the route discovery and maintenance phases efficiently. However, increased node mobility and node membership can lead to excessively high signalling traffic, leading to congestion and poor network performance. Intuitively, any routing protocol that tries to maintain state (e.g. a pre-computed source route, network topology) for routing purposes, appears non-scalable for ad hoc networks, since maintenance of the state requires additional signalling over the entire network.

Recent advances in positioning techniques such as GPS [7,8] and other ad hoc localization [9–12] has motivated researchers to pursue better routing schemes that take into account the physical location of the node (location is usually represented as a tuple consisting of latitude, longitude and altitude or as a point in space using cartesian coordinates). Many schemes have been proposed for position based routing in static ad hoc networks that guarantee loop free routes from a source to a destination [13–15]. One of the key observation from all of the position based routing algorithms is that the routing decision at an intermediate node is solely based on its position, its local information (position of neighbors), and the position of the destination. Nodes can periodically broadcast short packets containing their identities as well as locations so that each node is aware of its neighborhood. Another motivation for using geographic routing is that link breakages do not necessarily result in routes getting broken, since packets can be readily forwarded via alternate links, and are guaranteed to be delivered as long as a path exists in the network. Clearly, only localized algorithms provide scalable solutions, and geographic routing is indeed a potential candidate for scalable routing in a critically power/bandwidth constrained network.

Although geographic routing may be the key in providing a scalable solution for routing in ad hoc networks, an important requirement for any position based routing algorithm is the need for an accurate position information of the destination. Thus, a fundamental problem in geographic routing is maintaining the locations of nodes in a distributed manner in an ad hoc network such that the position of a required destination can be determined with minimal effort. Location management has been exhaustively studied in conjunction with Cellular networks, but the dynamic nature of the network as well as the scarcity of bandwidth makes this problem more interesting and challenging in ad hoc networks.

In this work, we describe a novel yet simple location management scheme for large ad hoc networks by using the notion of multilevel hierarchies. While the concept of hierarchical location/mobility management is not entirely new and has been commonly employed in Wide Area Networks [16], the contribution of this work lies mainly in specifying a distributed scheme suitable for ad hoc networks and showing that such a scheme scales very well for large mobile ad hoc networks. The design of our protocol is motivated by the following network characteristics:

(1) *Localized node mobility*: Often, node movement is limited to a specific locale in the network. There may only be few occasions when a node needs to traverse the entire network diameter.
(2) *Varying node mobility*: A network that has randomly distributed velocity profile consists of nodes that are highly mobile, slow and stationary.
(3) *Localized communication needs*: Communication requirements usually arise between nodes that are geographically separated by a small number of radio hops.

Under these observations, we note that an efficient location management protocol should recruit distributed location servers in such a way that location updates from mobile nodes update location databases in a timely manner with minimal resource usage. Since any signalling in a wireless ad hoc network consumes network bandwidth, the protocol must take into consideration the motion profile of each node so that a rapidly moving node may be required to transmit frequent updates while a static node may update less frequently. Finally, there must be an efficient balance between the update and paging process so that the location queries for nearby nodes stay local, while those for nodes that are located far away may incur additional cost.

The remainder of this paper is organized as follows: Section 2 describes the grid ordering scheme and details of the hierarchical grid location management protocol. In Section 3, we show that the average per node location update cost grows only logarithmically in the total number of nodes in the network for a uniformly randomly distributed network. Section 4 describes in detail the simulation environment and the parameters for our performance evaluation of our protocol against two well known location management schemes. We comment on the performance results obtained via simulations in section 5. Other related work is described in section 6. Finally, the distinct features of our contribution and concluding remarks are given in Section 7.

## 2 Hierarchical Grid Location Management (HGRID)

Clustering and hierarchical network organization techniques have been previously studied in multi–hop wireless networks for routing scalability [17–20]. However, the original motivation behind hierarchical routing is to reduce the memory requirements for storing routing table in internetworks of thousands of networks consisting of millions of hosts, and not to withstand the unpredictable and dynamic nature of mobile ad hoc networks. Although there are well known schemes to form unique clusters within the network and organize them into a hierarchical network, maintenance of this hierarchy with mobility would be extremely difficult, given the dynamic topology changes and the limited bandwidth by which the reconfiguration information must be propagated across the nodes. Cluster head changes can cause a *rippling effect*, where the role change of a leader in the higher level of hierarchy results in subsequent changes in lower leaders all the way to the bottom of the hierarchy. Another difficult task in maintaining a hierarchical network is the assignment and binding of node addresses to hierarchical addresses, which is primarily used for routing in the hierarchy. Periodic or triggered registration/binding of addresses to hierarchical addresses, and the query–response for obtaining the hierarchical address before the start of a communication session add to the control overhead, and hence, congests the network. Thus, if we were to design our location management protocol based on node membership, we would undoubtedly be facing a daunting task. But, if we are able to design a hierarchy based on unit regions, where the position of a node automatically assigns it a specific role in the hierarchy, we can get rid of complex message exchanges required to initialize and maintain a hierarchy across the network. This observation is precisely the motivation behind the HGRID protocol.

### 2.1 Hierarchical Grid Ordering

In order to partition the network into a hierarchy, we first divide the topography into unit square regions of side $\frac{r_t}{\sqrt{(2)}}$ (hereafter called $L_0$), where $r_t$ is the transmission range of a node. Such a design ensures that any two nodes within an $L_0$ grid can directly communicate with each other. Thus, designing a protocol that enables a mobile node to be aware of its neighbors in a $L_0$ grid is trivial (for e.g., a periodic broadcast protocol). The grid hierarchy in HGRID is defined by a recursive process as follows: at each level $i(1 \leq i \leq k-1)$, we select the top rightmost $L_{i-1}$ leader to be the $i^{th}$ hierarchical leader of the bottom left $L_i$ grid, top leftmost $L_{i-1}$ leader to be the hierarchical leader of the bottom right $L_i$ grid, bottom rightmost $L_{i-1}$ leader to be the hierarchical leader of the top left $L_i$ grid and bottom leftmost $L_{i-1}$ leader to be the hierarchical leader of the top right $L_i$ grid. The top of the hierarchy, $(L_k)$, is defined by the four $L_{k-1}$ grids. Note that this may not be the only way to form a hierarchy using unit grid $(L_0)$ regions. Certainly, other grid orderings are possible, but the general idea behind location management holds for all practical purposes.

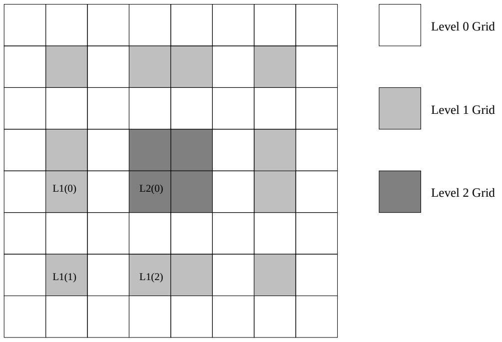


Fig. 1. A three level hierarchy in HGRID. A level 1 grid leader knows the exact location of all nodes located in the four level 0 grids under it. Level 2 leaders are constituted from level 1 leaders. A local broadcast protocol ensures that all the leaders in level 2 grids are aware of all the nodes in the network.

### *2.2 Location Management in HGRID*

Each time a node $u$ crosses an $L_0$ grid boundary, it broadcasts its entry into the new region, and unicasts two *LOC_UPDATE* (location update) packets – one to the $L_1$ grid of its previous $L_0$ grid (if required) indicating its departure from the region, and another packet to the $L_1$ grid of its current $L_0$ grid to indicate its arrival. Each packet contains information regarding the node's current and previous location, as well as the action to be taken (insertion/deletion from the location database) at each leader to make the location servers consistent in their view of the network. The *LOC_UPDATE* packets are processed at each level of the hierarchy in the following manner: node $v$ which is present in the hierarchical leader grid to receive the *LOC_UPDATE* first, updates its own location database, and broadcasts the packet in the current grid. Every location server that receives the broadcast simply updates its location database, if it is co-located in the same grid. Node $v$ also checks the *LOC_UPDATE* to see if the boundary crossing requires its hierarchical leader to be alerted, and if so, unicasts the packet to its next destination by geographic forwarding. If the movement specified in the *LOC_UPDATE* is within the area covered by the current hierarchical leader, $v$ decides to stop the registration process. Thus the location registration process continues until the *LOC_UPDATE* reaches either any one of the four $L_{k-1}$ leaders, or a hierarchical leader grid which covers the grid boundary whose crossing started the registration process. When a *LOC_UPDATE* reaches a $L_{k-1}$ leader, the node receiving the packet first carries out a local broadcast protocol to make all the $L_{k-1}$ leader databases consistent. Also, node $u$ carries out a location maintenance process similar to the previous approaches to update its location database to be consistent with others servers in its new grid.

**Lemma 1** *A LOC_UPDATE which was produced by an* $L_i^{th}$ *boundary crossing by a node visits at most i+1 hierarchical leaders for* $(0 \leq i \leq k-1)$ *and i leaders for* $i = k$ *in any* $L_{k-1}$ *grid.*

**PROOF.** Since by construction, $L_{i+1}$ leaders are constituted from $L_i$ leaders, any boundary crossing in an $L_{i+1}$ grid requires only the leaders from the lowermost level to the $i+1^{th}$ leader to be notified. Since there are at most k hierarchical leaders to be visited from the lowest level to the highest, an $L_k$ boundary crossing visits at most k leaders.

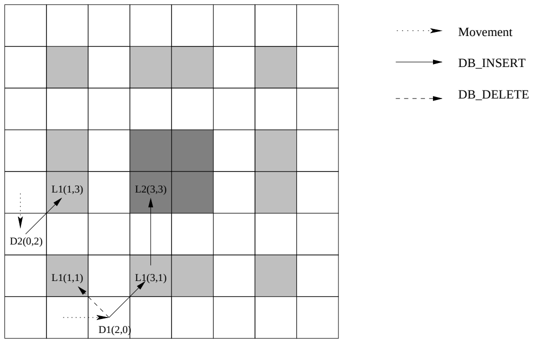


Fig. 2. Location Update in HGRID

**Example 2** *Figure 2 shows a typical location update process in the hierarchical location management scheme. Since* $D_1$ *crosses a level-1 boundary, it has to unicast two LOC_UPDATE packets, one to* $L_1$*(1,1), the* $L_1$ *leader of the previous* $L_0$ *grid it had visited, and the other to* $L_1$*(3,1), the* $L_1$ *leader of it's current grid. The update to* $L_1$*(1,1) instructs location servers within the grid to delete the entry for* $D_1$ *from their location databases (DB_DELETE), and the update to* $L_1$*(3,1) instructs the servers in* $L_1$*(3,1) to make a new entry for* $D_1$ *(DB_INSERT). The update to* $L_1$*(1,1) stops at* $L_1$*(1,1), while the update to* $L_1$*(3,1) continues to be unicast to* $L_2$*(3,3), since a level-1 boundary crossing requires the next hierarchical leader* $L_2$*(3,3) to be updated. Notice that the movement of another node* $D_2$ *terminates at* $L_1$*(1,3), since the movement is local to the* $L_1$ *hierarchy managed by* $L_1$*(1,3). Note that each step shown in the figure may consist of multiple physical transmissions of the update packet to reach the specified grid.*

**Location Discovery**: If destination $D$ is in the same grid as source $S$, $D$ would be in $S$'s neighbor table because of the local broadcast protocol. Otherwise, a *LOC_QUERY* (location query) packet for $D$ is sent to $S$'s $L_1$ leader. As part of location server set up, it is trivial to realize that if $S$ and $D$ are in the same $L_i^{th}$ grid, the query has to be forwarded until it reaches an $L_i^{th}$ server (in the worst case), before a location reply can sent back. Since the location databases in the upper levels of the hierarchy carry the approximate location information of nodes, location replies from these servers return the address of the server who has more accurate information of the destination.

**Lemma 3** *A LOC_QUERY visits at most k-1 hierarchical leaders.*

**PROOF.** Since the query has to be forwarded to a $L_{k-1}$ leader in the worst case, and since the servers in the top level have complete knowledge about the network, the lemma follows.

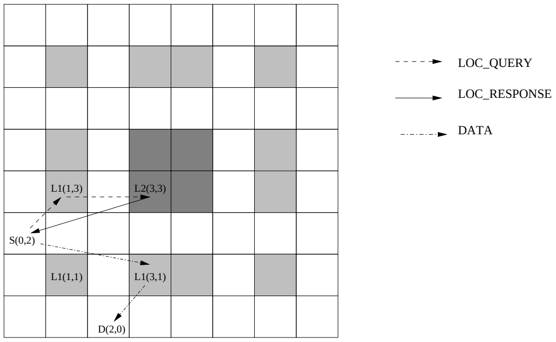


Fig. 3. Location Discovery and Data Transfer

**Example 4** *In figure 3, the location query from source S is forwarded by* $L_1$*(1,3) to* $L_2$*(3,3), since it has no knowledge of destination D. The query terminates at* $L_2$*(3,3), and a location reply is returned to S with the destination location specified as* $L_1$*(3,1) in the reply packet.*

### *2.3 Data Transfer*

If $S$ and $D$ are co–located in the same $L_1$ grid, the location of $D$, as indicated by the location reply, is accurate and $S$ can forward the data directly to $D$'s location. Otherwise, the location is approximate, and $S$ forwards the data packet to the location server specified in the location reply. When the data packet reaches the specified grid, a server $v$ that receives the packet first checks its neighbor table to see if $D$ is co-located in the same grid. If so, the packet is successfully forwarded to the destination. Otherwise, $v$ searches for the $D$ in its location database. By construction, $v$ must have an entry for $D$ in its location database. If $v$ has accurate information about $D$, it further forwards the packet to $D$, otherwise, the packet is forwarded to the next location server which is a level lower than $v$ in hierarchy, but has more accurate information about $D$'s position. This process continues until the packet reaches $D$, or it reaches an $L_0$ grid, and the node that receives the packet drops it since it has no information about $D$. This can happen because $D$ would have left this grid, and $D$'s location update to its new hierarchical leader failed to reach the leader before the data packet was forwarded by the leader to $D$'s previously visited grid. Since packet transmission time is much smaller than node mobility, such drops should not be frequent under low network load conditions.

**Example 5** *In figure 3, S forwards the data packet to $L_1$(3,1), as indicated by the location response from the location server. When the packet reaches $L_1$(3,1), the node that receives the packet searches its location database, and realizes that D is located in (2, 0). It then forwards the packet to (2, 0) which is successfully received by D.*

Thus, under the HGRID protocol, routing may be sub-optimal due to the fact that (i) the location servers have partial information regarding the nodes in the network, and (ii) location replies may carry approximate node locations. The first data packet to arrive at the destination may suffer by taking a much longer route than the shortest path to the destination, had the precise location of the destination been known. To alleviate this problem, when $D$ receives a packet from $S$, it can send a notification to $S$ with its current location so that the new packets to $D$ do not suffer the same fate as their predecessor. For TCP connections, this can easily be built into the acknowledgement packet that $D$ proceeds to send to $S$.

## 3 Location Update Cost Analysis

We assume a uniformly randomly distributed $N$ node network in a square region of area $A$ in which nodes move randomly and independent of each other. Each node selects a direction to move, chosen uniformly between $[0, 2\pi]$. Each node also selects its speed, chosen uniformly between $[v - c, v + c]$ for some average velocity $v$ and constant $c$ for some time $t$, where $t$ is distributed exponentially with mean $\tau$. After a node has traveled for time $t$, it selects another direction, speed, and time to travel. As a consequence of this model, the average degree of a node will be proportional to $\pi r_t^2 N/A$ where $r_t$ is the radio range and $\pi r_t^2$ is the area within a node's transmission range. To keep this fraction constant, $A$ must grow linearly with $N$.

Woo and Singh [21] were the first to analyze the scalability of a location management protocol in a theoretical framework. Under this framework, the cost of a location management protocol is analyzed by using a specific mobility model and a specific geographic routing algorithm, as the *average number of packets* sent within the network in order to maintain the locations of the nodes. The main observations from [21] are the following:

1. *The cost of broadcasting, $b$, in an unit square by a node is proportional to the number of transmissions needed to cover the said region. The latter is in turn proportional to the area of the unit square $a$ divided by the area covered by a single transmission. Thus, $b = O(a/r_t^2)$ packets per unit square.*
2. *The distance a node has to cover to cross an unit square is proportional to the side of an unit square. Thus, the number of unit squares a node crosses per second, $\rho$, is proportional to $v/\sqrt{a}$.*
3. *Given a source-destination separated by euclidean distance $d$, the number of transmissions that need to be carried out to send a packet from the source to destination is given by $d/z$, where $z$ is the average forward progress made in the course of one transmission.*

Note that from [22], it is known that $z$ can be computed from $r_t$ and the average degree of a node in the network. With these observations, we are ready to calculate the average location update cost of the HGRID protocol.

Let $D$ be the average distance traversed by *LOC_UPDATE*s for any boundary crossing and let $\hat{d_i}$ be the total average distance traversed by both *LOC_UPDATE*s on an a $L_i^{th}$ boundary crossing. Since $\hat{d_i}$ can be at most twice the maximum average distance traversed by any one of the *LOC_UPDATE*s on an $L_i^{th}$ boundary crossing, we have:

$$\hat{d_i} \leq 2 \times \bar{d_i} \tag{1}$$

Here we take maximum average distance to be the distance traversed by the *LOC_UPDATE* from the current $L_0$ grid to the $i^{th}$ hierarchy.

For the average distance traversed as a result of boundary crossing in each level of the hierarchy, we have:

$$\begin{aligned}
\bar{d_1} &\leq r_t(\frac{2+\sqrt(2)}{4}) \\
\bar{d_2} &\leq \bar{d_1} + 2 \times r_t(\frac{2+\sqrt(2)}{4}) \\
&\leq 3 \times r_t(\frac{2+\sqrt(2)}{4})
\end{aligned}$$

In general, the average distance traversed in the $i^{th}$ hierarchical level is

$$\bar{d_i} \leq 2^{i-1} \times r_t(\frac{2+\sqrt(2)}{4}) \qquad (1 \leq i \leq k-1)$$

$\bar{d_k}$ is a special case, since in the best case, we have to broadcast only once (the $L_k^{th}$ boundary crossing into a topmost leader), and in the worst case, we have to visit *k-1* leaders starting from $L_0$. Therefore

$$\begin{aligned}\bar{d_k} &\leq r_t(\frac{1+2+4+...+2^{k-2}}{2}) \\ &\leq r_t(\frac{2^{k-1}-1}{2}) \\ &\leq 2^{k-1} \times r_t(\frac{2+\sqrt(2)}{4})\end{aligned}$$

Generalizing, the average distance traversed by a *LOC_UPDATE* in the $i^{th}$ hierarchical level is

$$\bar{d_i} \leq 2^{i-1} \times r_t(\frac{2+\sqrt(2)}{4}) \qquad (1 \leq i \leq k) \tag{2}$$

In addition, $P_i$, the probability that the $L_i^{th}$ server is updated, is:

$$\begin{aligned}P_1 &= \frac{4^k}{2 \times 4^k \times [1-2^{-k}]} \\ &= \frac{1}{2 \times [1-2^{-k}]} \\ P_2 &= \frac{\frac{4^k}{2}}{2 \times 4^k \times [1-2^{-k}]} \\ &= \frac{P_1}{2}\end{aligned}$$

In general, the probability values are

$$P_i = \frac{P_1}{2^{i-1}} \qquad (1 \leq i \leq k) \tag{3}$$

Average distance traversed by a *LOC_UPDATE*

$$\begin{aligned}D &= \sum_{i=1}^{k} P_i \hat{d_i} \\ &= 2 \times \sum_{i=1}^{k} P_i \bar{d_i}\end{aligned}$$

$$
\begin{aligned}
&\leq 2P_1\bar{d}_1 \times \sum_{i=1}^{k} \frac{2^i-1}{2^{i-1}} \\
&\leq 2r_t(\frac{2+\sqrt(2)}{4})(\frac{k}{1-2^{-k}}-1) \\
&= O(k) \text{ for large } k \qquad (4)
\end{aligned}
$$

A location update cost consists of $\frac{\bar{d}_i}{z}$ unicasts and $i$ broadcasts (by *lemma 3.1*) for a $L_{i-1}^{th}$ boundary crossing. The average number of broadcasts

$$
\begin{aligned}
\bar{b} &= \sum_{i=1}^{k} iP_i \\
&= 2 - \frac{k}{2^k-1} \\
&= O(1) \text{ for large } k \qquad (5)
\end{aligned}
$$

Hence, average location update cost per node for the hierarchical grid location management is

$$
\begin{aligned}
c_u &= \rho\ (\frac{D}{z} + \bar{b}) \\
&= \rho\ (\frac{D}{z} + O(1)) \\
&= \rho\ O(\frac{k}{z}) \\
&= O(\rho\ log_2 N) \\
&= O(v\ log_2 N) \text{ packets/sec/node} \qquad (6)
\end{aligned}
$$

Thus, the average per node location update cost in HGRID grows only logarithmically in the total number of nodes in the network and is proportional to the average speed of nodes in the network.

## 4 Performance Studies

We implemented the HGRID protocol as well as SLaLoM [23] and SLURP[21], two well known protocols described in literature, in Glomosim [24]. A location management layer was built in to the TCP/IP protocol stack that operated in conjunction with IP as the network layer protocol. Main data structures in the location management layer consist of i) a *location table* and ii) a *neighbor table.* When a location server node receives a location update packet from a node, the current location of that node is updated in the location table. A periodic broadcast protocol enables each node to realize its local connectivity, and records it in the neighbor table to assist in geographic routing. Apart from having a location database, nodes are also equipped with a *live connections* table, which is updated when a node receives a data packet or a location change notification. Location change notifications are sent out by end points of a data connection when their locations change significantly from their previously advertised locations. This table reduces the number of query packets transmitted during a session, since later data packets in a session can use the destination's location entry in the table until the entry becomes invalid due to a timeout. The timeout is determined by the average time it takes a node to move out of a unit grid with an average velocity specified by its mobility. Data from the transport layer is queued in a packet buffer if the location of the destination is unknown, and a location query packet is transmitted to the destination's location server. Packet lifetime in the buffer is 4 seconds, and is subsequently dropped if a location query sent out fails to return the location of the destination within this time. MFR [25] without backward progression, in which packets are dropped if no forward progress can be made, was implemented as the geographic routing algorithm.

Motivated to study the performance of the protocols for network scalability as well as robustness against node mobility, we carried out two sets of simulations. The first study fixed the node density to be 80 $nodes/km^2$ and the average node speed to be 5 $m/sec$ while varying the total number of nodes in the network. The second study evaluated all the protocols in a square topography of $2000 \times 2000$ consisting of 320 nodes by varying the average node speed. Specific parameters for our simulations are listed in table 1.

Node home regions in SLaLoM and SLURP are randomly assigned using a simple hash function that maps the IP address of the node to one of the unit squares. Since the performance of SLALoM is dependant on the selection of $K$, we decided to vary $K$ from 2 TO 6 in the first study, and to simulate two versions of SLALoM, with $K = 2$ and $K = 4$ (hereafter called SLALoM-$K_2$ and SLALoM-$K_4$ respectively), for the second study. For the first study, HGRID defines up to 5 hierarchical levels, and a 3 level hierarchy for the second. The different packet types and location management overhead in bytes for all protocols are given in table 2.

In order to test the network under stable conditions, we let the nodes move around for the first 150 seconds of the simulation, so that location server and database set up is initialized appropriately and the control traffic is stabilized. To test the efficiency of the protocols for correct location discovery as well as efficient delivery of data, we initialized 1000 CBR connections for the first sce-

Table 1
Simulation Parameters

| Parameters | Scenario I | Scenario II |
|---|---|---|
| Simulation Time | 300 sec | 300 sec |
| Simulation Area | 1000x1000m - 6000x6000m | 2000x2000m |
| Unit Grid Size | 250m | 250m |
| Number of Nodes | 80 - 2880 | 320 |
| Node Density | 80 nodes/$km^2$ | 80 nodes/$km^2$ |
| Transmission Range | 350m | 350m |
| Transmission Speed | 54 Mbps | 54 Mbps |
| MAC Protocol | IEEE 802.11g | IEEE 802.11g |
| Mobility Model | Random Waypoint | Random Waypoint |
| Maximum Speed | 10 m/sec | 0-25 m/sec |
| Minimum Speed | 0 m/sec | 0 m/sec |
| Pause Time | 0 sec | 0 sec |
| CBR connections | 1000 | 160 |
| CBR Rate | 1 packet | 2 packets/sec |
| Data Payload | 512 bits | 512 bits |
| Traffic Pattern | Random | Random |
| Buffer Size | 1000 packets | 1000 packets |

Table 2
Packet types and overhead

| Packet Type | SLURP | SLALoM | HGRID |
|---|---|---|---|
| Update | 33 | 35 | 34 |
| Query | 37 | 37 | 37 |
| Response | 53 | 53 | 53 |
| Notification | 33 | 33 | 33 |
| Maintenance | 33+20n | 33+20n | 33+20n |

nario. The source of each connection attempts to send exactly *one* data packet by initiating a query to locate the destination, and sends the data packet to the location returned in the reply packet. A connection is declared successful if the data packet can be correctly received by the destination. For the second study, we imposed high traffic on the network with 50% of the nodes initiating UDP connections, each session having a rate of 2 packets/second. The source and destination nodes are chosen randomly for both the scenarios, randomly starting after 150 seconds into the simulation and terminating randomly at 250 seconds into the simulation.

## 5 Simulation Results

### *5.1 Scalability with Network Size*

The results shown in this section show the effect of increasing average path length and network size on location management and geographic routing. An increase in the average path length proportionally affects the time taken to update location servers, and an increase in network size results in a high volume of updates to servers. We study the effect of this phenomenon on the performance of the network. Each plot point presented here is an average of seven simulation runs.

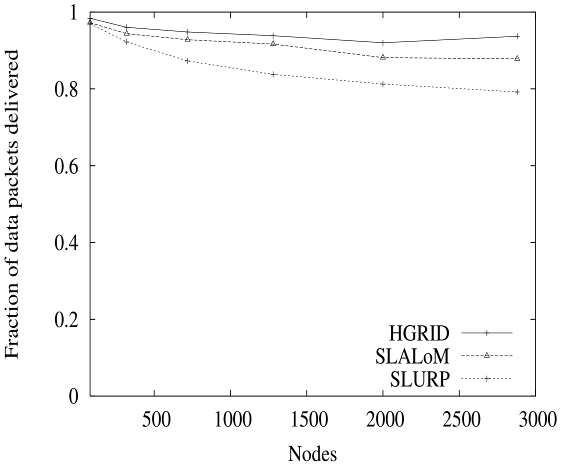


Fig. 4. Data Throughput

Figure 4 shows the fraction of data packets successfully delivered by each protocol. HGRID is able to deliver packets in more than 90% of the cases for all scenarios, and hence performs best. SLURP performs worst, with significant performance degradation for network sizes of 1500 nodes and more. The under performance of SLURP can be attributed to the higher percentage of queries that fail to return the location of the destination and higher delay of location responses. Recall that each data packet is held for only 4 seconds in the buffer, and if a query–response fails to terminate within this period, the packet is dropped. Hence, the delayed responses in SLURP are useless since the packets awaiting the destination's location would already have been discarded at the source.

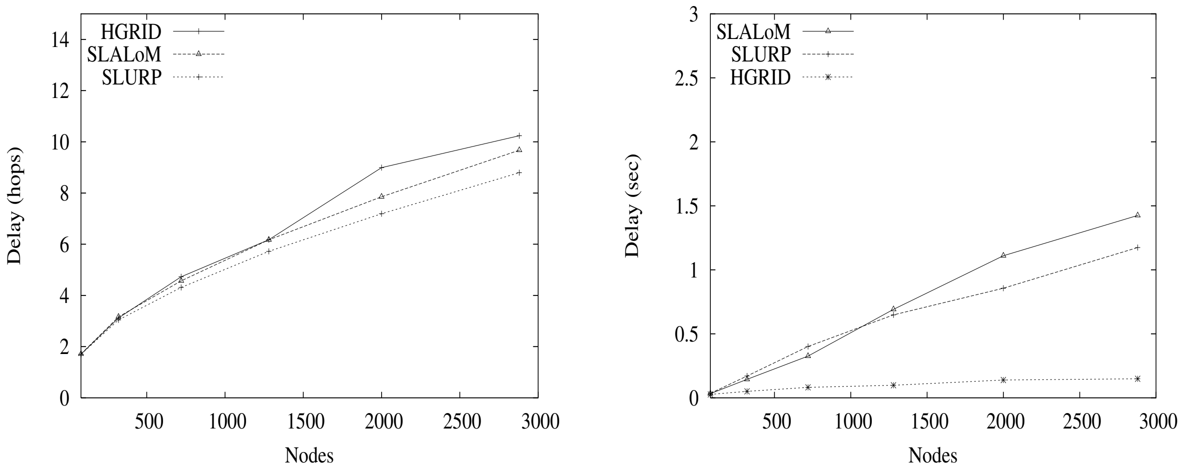


Fig. 5. Average end-to-end data delay

Figures 5(a) and 5(b) show the delay experienced by successfully delivered data packets in number of hops (transmissions) and seconds. Since the location responses convey the exact location of the destination in SLURP, packets take the shortest path in SLURP. HGRID and SLALoM take longer paths, since on average, packets are routed to near home regions in SLALoM or hierarchical location servers having more accurate knowledge of the destination in HGRID, before being handed over to the destinations finally. However, the average delay (time) experienced by data packets follows a different trend as shown by figure 5(b). Even though packets take the longest paths in HGRID, they have the lowest average delay, since the network is least congested. Since higher control overhead adversely affects data on the shared channel, a lower data delay in SLURP indicates that the network is less congested in SLURP than SLALoM, and is easily verified by figure 7.

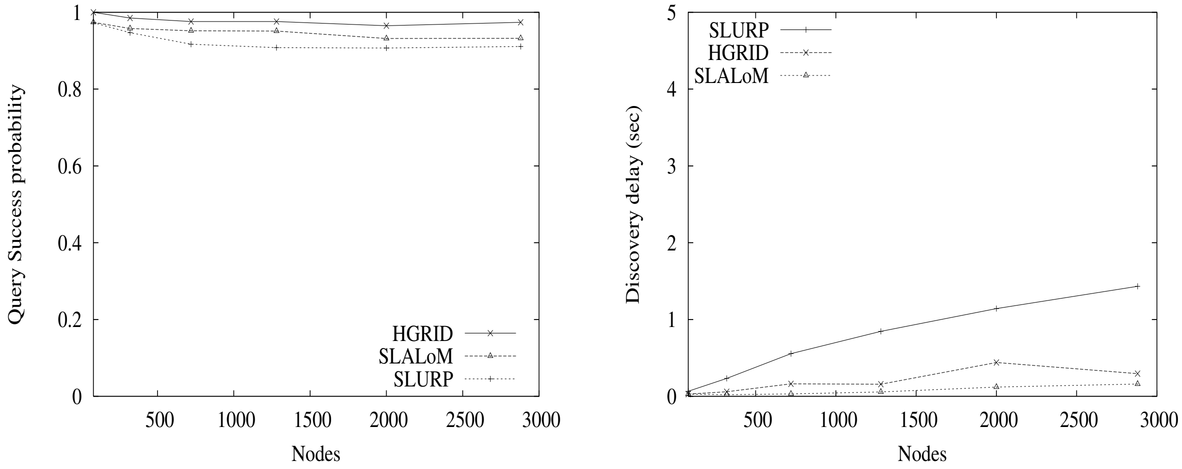


Fig. 6. Location discovery probability and delay

Figure 6 shows the query success probability and the average delay for location discovery for the three protocols. All protocols perform well, with more than 90% of the queries returning the location of the queried destination. Packets can be dropped due to two reasons in our simulations – *geographical holes* [13] and IEEE802.11 induced congestion. Since the home region of an arbitrary node is located furthest from an enquirer in SLURP (since home regions are chosen randomly, there is no guarantee that the region is actually close to the enquirer) compared to the other protocols, the location discovery takes longest to complete in SLURP. Also, since the home region is located further, this increases the dropping probability of a query or response packet, and thus decreases the query success probability. SLALoM has the closest home region for any node for the scenarios considered, and has the lowest delay for location discovery. However, high network congestion due to the higher control overhead causes additional packet drops in SLALoM, and thus has lower query success probability than HGRID.

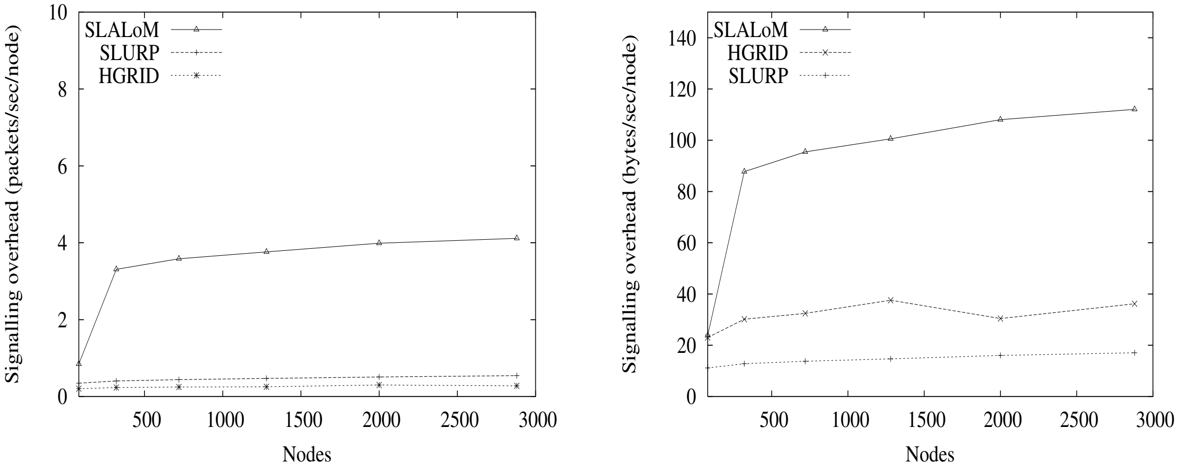


Fig. 7. Average control overhead

Figures 7(a) shows the per node average control overhead in packets per unit time for all the protocols. With the number of CBR sessions being low, most of the signalling overhead is due to update traffic. As indicated by the analysis, HGRID has the least control overhead, since the update overhead increases only logarithmically with the number of nodes. For practical scenarios such as the ones considered in our simulations, SLURP performs better than SLALoM with respect to control overhead. Also, SLURP performs best when one considers the overhead in bytes instead of number of packets, as shown by 7(b). With the packet overhead being nearly equal for all the protocols, the higher overhead for HGRID and SLALoM can be explained by the overhead incurred in location maintenance. The average number of location entries that have to be transmitted is more for SLALoM and HGRID (see figure 8), resulting in additional bytes being transmitted in these two protocols.

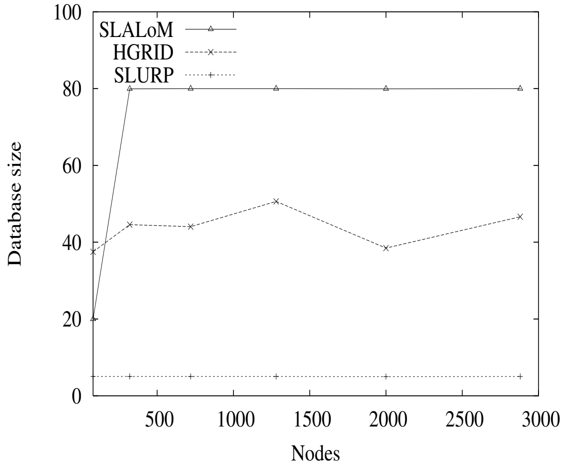


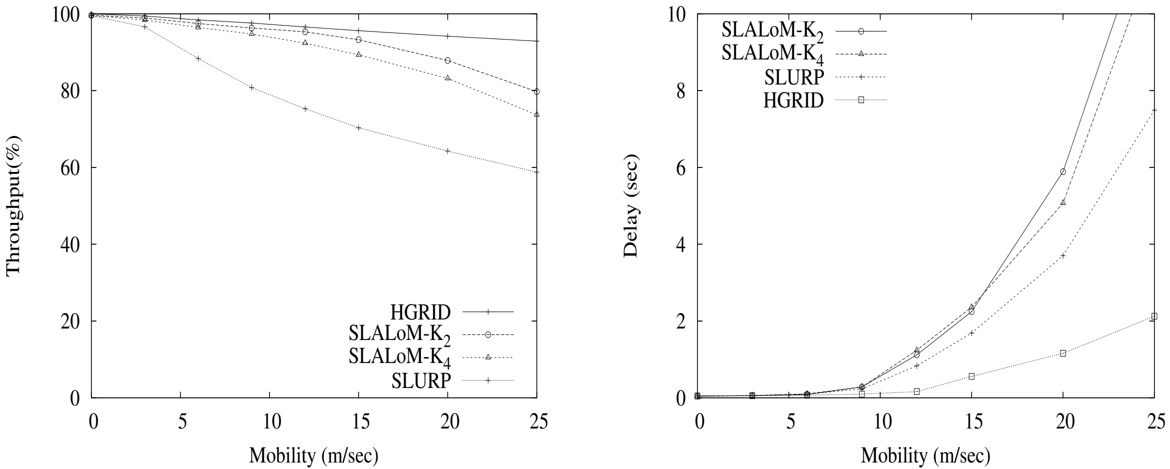
Fig. 8. Average location database size

Finally, figure 8 shows the increase in location database size of each protocol with increase in number of nodes. SLURP has minimal memory requirements, since each node can choose any unit region randomly from the available set of Order–1 region as its home region. In SLALoM, the set of available Order–1 region is restricted to only $K^2$. While lower order servers have a few location entries, the higher order servers have to retain location information about almost all nodes in HGRID. However, the average of these grows only slightly with the increase in network size. Thus, all protocols are scalable, with the database size being nearly constant or increasing marginally with network size.

### *5.2 Scalability with Mobility*

The results shown in this section show the effect of mobility on location management and how different location management protocols affect the performance of geographic routing. We varied the maximum speed in the Random Waypoint model to change the average mobility of the nodes. An increase in mobility is proportional to the rate at which nodes cross grid boundaries, and hence the rate at which new updates are sent out to location servers. We study the effect of this phenomenon on the performance of the network. Each plot point presented here is an average of seven simulation runs.

Figures 9(a) and 9(b) show the average data throughput and delay achieved by each protocol. Throughput decreases with mobility for all the protocols, with SLURP being affected most by mobility, and HGRID performing best. HGRID gives a near steady performance, delivering a throughput above 90% in all the cases. SLALoM-$K_2$ performs slightly better than SLALoM-$K_4$. Packet delay increases with mobility, with SLALoM-$K_2$ performing worst, indicating that network congestion due to mobility causes network under–performance, and that the rate at which updates are sent out affect each protocol with different



(a) Throughput (b) Delay

Fig. 9. Average data throughput and end-to-end delay

degree. Since control overhead is least for HGRID (see fig. 11(a)), this easily explains why HGRID performs best. Recall that SLALoM groups $K^2$ Order-1 regions to form Order-2 regions. Thus, in our simulations, each node in SLALoM-$K_2$ has 16 home regions, both near and far, while nodes in SLALoM-$K_4$ only have 4 home regions, all being near. Hence, a boundary crossing in results in more updates in SLALoM-$K_2$ than SLALoM-$K_4$. On average, both overheads would be more than that of SLURP, since each node has exactly one home region in SLURP. Since higher control overhead adversely affects data on the shared channel, one would expect better performance for SLURP than SLALoM. This can be explained by figures 10(a) and 10(b).

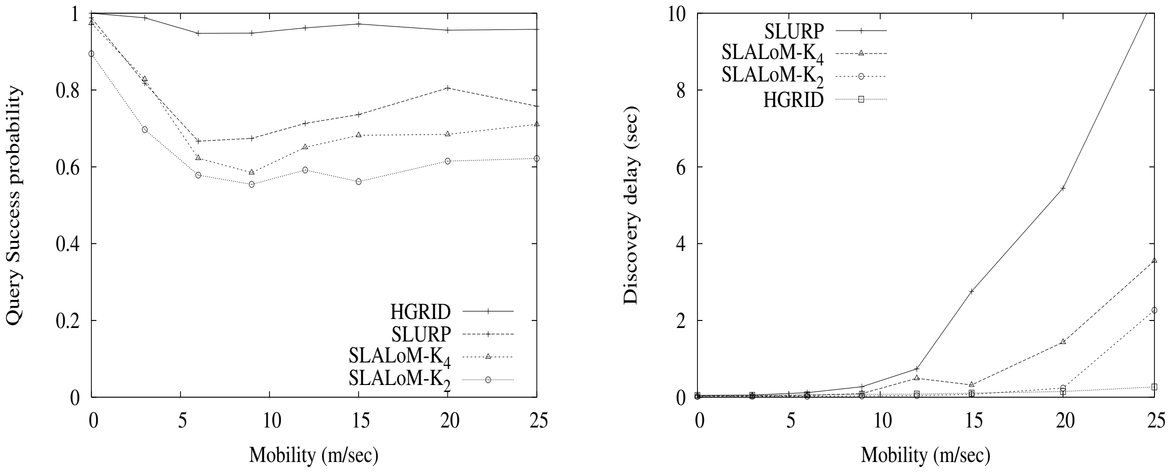


Fig. 10. Location discovery probability and delay

Figure 10(a) shows the probability that a query for a destination returned successfully, and figure 10(b) shows the average delay for location discovery via the query-response phase. Since control overhead is least for HGRID, most queries are not affected by network congestion, and destinations are easily located within milliseconds. However, the high control overhead affects SLALoM-$K_2$ severely, with a lot of discovery packets being dropped due to MAC layer contention (a quick look at the number of CTS packets ignored by IEEE802.11 due to software carrier sense easily confirmed this). Although the home regions are located closer to the enquirer than that in SLALoM-$K_4$ or SLURP, the probability that a destination's location is discovered is smaller in SLALoM-$K_2$. This accounts for the reduced throughput for SLALoM-$K_2$. However, since the home region of an arbitrary node is located further from an enquirer in SLURP (since home regions are chosen randomly, there is no guarantee that the region is actually close to the enquirer), the location discovery takes longest to complete in SLURP. In the worst case, the query-response phase takes more than 10 seconds in SLURP. This behavior also explains why SLALoM-$K_2$ performs better than SLALoM-$K_4$ in terms of location discovery delay. Since data packets have a lifetime of only 4 seconds in the buffer, the delayed responses in SLURP are useless since the packets awaiting the destination's location would already have been discarded at the source. Delayed responses are more costly than no responses at all, as indicated by the poor throughput of SLURP. However, given the same number of hops for a data session, SLURP performs better than SLALoM in terms of packet delay, since higher congestion due to update traffic causes low priority data to be queued longer in SLALoM.

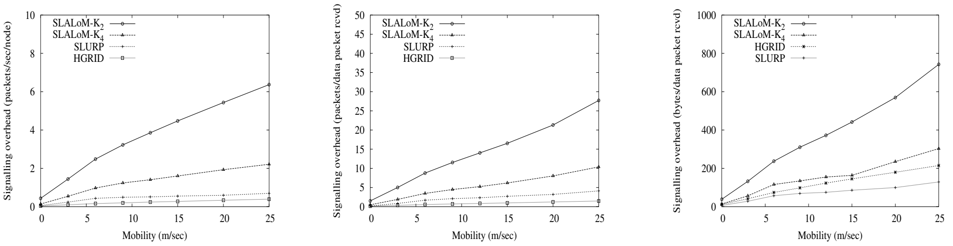


Fig. 11. Average control overhead

Figures 11(a) and 11(b) show the control overhead incurred by each protocol in number of packets. Clearly, the overhead due to update traffic increases with mobility for all protocols, and is highest in SLALoM because a grid crossing results in updating multiple home regions simultaneously. SLALoM-$K_4$ performs better than SLALoM-$K_2$ since it only has to update 4 home regions at a time. HGRID performs best, since updates are forwarded to higher level servers only when the movement is across a grid boundary that is not contained within the current hierarchical leader grid. Hence the number of updates are much smaller in HGRID than other protocols. However, this is no longer true if one considers the overhead in bytes than the number of packets per received data packet, as shown by figure 11(c). HGRID location server grids are concentrated towards the center of the terrain, and as the hierarchy of the server grid becomes higher, so does the number of location entries in the location database in a location server which is present in it. When a node moves into a server grid, it has to receive all the entries of nodes that are registered under that grid as part of database maintenance, and the number of such entries increases the overhead in location maintenance. Server grids become points of congestion and thus the bottleneck in performance as indicated by the increase in data discovery and data delay with increased mobility.

## 6 Related Work

Finn's *Cartesian routing* [26] is the earliest known position based routing mechanism found in literature for datagram routing in wired networks. Greedy forwarding, in which next hop neighbors for packet forwarding are selected by a greedy decision, has been an efficient contender for position based routing in wireless packet networks. Most Forward within Radius with backward progression (MFR) [25], Nearest Forward Progress (NFP) [22], Compass routing [27] etc., are examples of routing protocols that make use of greedy forwarding. A main disadvantage of greedy forwarding is the occurrence of a local maxima from which it may not recover. For e.g., in MFR, an intermediate node may not have any active neighbors that can make forward progress towards the destination, and will drop the packet. To counter this, Karp et. al. [13] suggested *Greedy Perimeter Stateless Routing* (GPSR) in which a packet follows the path according to the well known *right-hand* in planar graphs rule to traverse around the local maxima. Independently, Bose et. al. suggested the Face-II [14] algorithm using a similar concept to GPSR to route messages using the right hand rule in planar graphs. Excellent surveys on position based routing algorithms can be found in [28] and [29].

Most of the literature in position based routing assume the presence of a location service through which the position of the destination can be obtained, but neither considers the details of such a service nor provides an insight into the scalability of geographic routing imposed by this service. Routing protocols such as Location Aided Routing (LAR) [5] and Distance Routing Effect Algorithm for Mobility (DREAM) [6] are the earliest known schemes for location based routing in literature. Although these were not openly proposed for the location management problem, they can be easily modified to fit the bill. DREAM proactively disseminates node position information so that future data sessions may flood data packets in the direction of the previously known position of the destination. On the other hand, LAR tries to find a source route to the destination and controls the request flood region by restricting it to an an area that includes the previous known location of the destination.

However, both these protocols are not scalable, due to their flooding nature of control packets. Haas and Liang [30] suggesed the earliest known location management protocol described in literature for ad hoc networks, even though they do not specifically deal with geographic locations in their work. The idea here is to select a subset (quorum) of nodes for storing pertinent information (node locations, in our case). Updated information is written to a subset of these nodes. When the intersection of these subsets is non–empty, any of the subset can be read for data written to the subset during a previous write cycle. Since there can be multiple responses to a read request, time stamps can be used to separate the most recent information. *Home region* based location management was independently suggested by Woo and Singh [21] and Stojmenovic[31]. Location management in [21] is similar to that of Mobile IP, where each node selects a unit region in the terrain as its *home region* by using a mapping function (a hash function, for e.g.) and the unique address of the node, and updates the region when the position of a node changes significantly. Home regions can be later queried by source nodes in an on demand fashion for locating destination nodes. The authors also showed that the location management cost due to this protocol grows as $O(vN^{\frac{3}{2}})$. Cheng et. al. [23] later improved this bound to $O(vN^{\frac{4}{3}})$ using the concept of *near* and *far* home regions. Similar to the region based protocols, Jinyang Li et. al. [32] and Xue et. al. [33] suggested alternate grid ordering schemes for achieving scalability in location management. However, the authors did not give any bounds on the location management cost for their schemes.

## 7 Conclusion

In this paper, we have evaluated the performance of HGRID, a novel multilevel hierarchical scheme for location management in geographic routing networks. To the best of our knowledge, this is the first solution that addresses the location management problem in mobile ad hoc networks from a hierarchical perspective. In order to separate the control from overhead due to mobility induced hierarchy maintenance, we suggest the establishment of a hierarchy from locale and topography information. Thus, we have combined the strength of existing schemes to establish a tree-like grid ordering scheme that strikes an efficient balance between the location management primitives for low update overhead as well as quick location discovery. We have shown the location update overhead in HGRID grows only logarithmically in the total number of nodes in the network, which is a significant improvement over the location management overhead incurred by existing protocols. We have carried out extensive simulations to study the performance of the protocol for scenarios that could not be incorporated into the analysis. Our results show that HGRID scales very well with increasing network size and is highly resistant to network topology changes induced by node mobility.

Although the proposed scheme has distinct advantages, we note that the use of hierarchies puts unfair requirements (in terms of resources and energy) on nodes that are location servers and higher order leaders. This issue is inevitable in any hierarchical scheme, and can be alleviated by cyclic rotation of roles among all the participating grids temporally. Alternately, leader grids can be treated as *voids*, and data packets may be routed around them using routing protocols such as GPSR or Face-II to relieve them of some burden. Another problem that is particular to locale based protocols occurs when nodes move out of a region leaving it empty. We are currently working on a solution that will delegate a neighboring region to temporarily carry out the duties of the empty region, and restore the state back to normal when a new node enters the previously empty region. Finally, boundary based location update schemes result in frequent updates for nodes that move back and forth between boundaries, creating a state of uncertainty with respect to the actual location of the node. Local forwarding techniques, such as the one suggested in [34], can be used to significantly reduce the severity of this problem. Overall, considering the pros and cons of the proposed scheme, we feel that the HGRID protocol efficiently operates in conjunction with the geographic routing protocol for scalable routing in large mobile ad hoc networks.

## References

[1] S. Murthy and J. Garcia-Luna-Aceves, “A Routing Protocol for Packet-Radio Networks,” *Proceedings ACM/IEEE MobiCom*, November 1995.

[2] T. W. Chen and M. Gerla, “Global state routing: A new routing scheme for ad-hoc wireless networks,” *In Proceedings of IEEE ICC*, June 1998.

[3] J. Broch, D. B. Johnson, and D. A. Maltz, “Dynamic Source Routing in ad hoc wireless networks,” *Mobile Computing, Kluwer Academic Publishers*, pp. 153–181, 1996.

[4] C. E. Perkins and E. M. Royer, “Ad hoc On-Demand Distance Vector Routing,” *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, February 1999.

[5] Y. B. Ko and N. H. Vaidya, “Location-Aided Routing (LAR) in mobile Ad-Hoc networks,” *IEEE Infocom*, pp. 66–75, October 1998.

[6] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility (DREAM),” *Proceedings ACM/IEEE MobiCom*, pp. 76–84, 1998.

[7] P. Enge and P. Misra, “Special Issue on GPS: The Global Positioning System,” *Proceedings of the IEEE*, pp. 3–172, January 1997.

[8] I. A. Getting, "The global positioning system," *IEEE Spectrum*, December 1993.

[9] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications*, October 2000.

[10] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf based user location and tracking system," *IEEE Infocom*, March 2000.

[11] A. Chakraborty and H. Balakrishnan, "The cricket location support system," *ACM Mobicom*, August 2000.

[12] L. Doherty, L. E. Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," *IEEE Infocom*, April 2001.

[13] B. Karp and H. T. Kung, "GPSR : Greedy perimeter stateless routing for wireless networks," *Proceedings ACM/IEEE MobiCom*, pp. 243–254, August 2000.

[14] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[15] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Communications*, vol. 8, pp. 48–57, February 2001.

[16] F Kamoun and L Kleinrock, "Stochastic Performance Evaluation of Hierarchical Routing for Large Networks," *Computer Networks*, vol. 3, pp. 337–353, November 1979.

[17] G. Pei, M. Gerla, and X. Hong, "Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility," *Proceedings of IEEE/ACM MobiHOC 2000*, pp. 11–18, August 2000.

[18] G. Pei and M. Gerla, "Mobility management for hierarchical wireless networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 6, no. 4, pp. 331–337, 2001.

[19] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable routing strategies for ad-hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1369–1379, August 1999.

[20] E. M. Belding-Royer, "Multi-level hierarchies for scalable ad hoc routing," *Wireless Networks*, vol. 9, no. 5, pp. 461–478, September 2003.

[21] S. Woo and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 7, pp. 513–529, January 2001.

[22] T. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 1, no. 34, pp. 38–44, 1986.

[23] C. Cheng, H. Lemberg, S. Philip, E. van den Berg, and T. Zhang, "SLALoM: A Scalable Location Management Scheme for Large Mobile Ad-hoc Networks," *Proceedings of Wireless Communications and Networking Conference*, March 2002.

[24] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," *Proceedings of the 12th Workshop on Parallel and Distributed Simulations – PADS '98*, May 1998.

[25] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–257, 1984.

[26] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," *ISI Research Report ISU/RR-87-180*, March 1987.

[27] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," *Proc. 11 th Canadian Conference on Computational Geometry*, pp. 51–54, 1999.

[28] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, no. 15(6), pp. 30–39, November 2001.

[29] I. Stojmenovic, "Position based routing in ad hoc networks," *IEEE Commmunications Magazine*, vol. 40, no. 7, pp. 128–134, July 2002.

[30] Z. J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 228–240, April 1999.

[31] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks," *Advances in Information Science and Soft Computing, WSEAS Press*, pp. 6–11, 2002.

[32] J. Li, J. Janotti, D. S. J. D. Couto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proceedings ACM/IEEE MobiCom*, pp. 120–130, August 2000.

[33] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad-hoc networks," *Proc. of the IEEE Conference on Local Computer Networks*, 2001.

[34] S. Philip and C. Qiao, "Elf: Efficient location forwarding in ad hoc networks," *Proceedings of the IEEE Globecom*, 2003.