

# Performance Evaluation of Location Management Protocols for Mobile Ad hoc Networks

**Sumesh J. Philip**

Department of Computer Science  
Western Illinois University  
Macomb IL 61455  
Email: sj-philip@wiu.edu

**Joy Ghosh, Swapnil Khedekar, Chunming Qiao**

Department of Computer Science and Engineering  
SUNY at Buffalo  
Amherst NY 14260  
Email: {joyghosh,khedekar,qiao}@cse.buffalo.edu

**Abstract** — *With the availability of location based services, applications that use locations for optimal performance will require efficient location management algorithms, where user locations are kept track with minimal system overhead. Geography based routing in mobile ad hoc networks is one such application that uses location information of nodes in a network to route data packets. Previous work in this area has shown that the selection of location management protocol is critical to the performance of such routing algorithms. Many location management schemes have been proposed in literature, and in an effort to quantitatively compare the performance of some of these schemes, we carry out extensive simulations to study SLURP, SLALoM and HGRID, three grid based protocols described in literature. Our study is two pronged – to compare the performance of location management with network mobility as well as traffic load, and to evaluate the effect these protocols have on the performance of geographic routing in mobile ad hoc networks. Our results show that the Hierarchical Grid Location Management protocol (HGRID) achieves steady performance for data throughput and delay, and minimally affects the performance of geographic routing.*

**Keywords:** Ad hoc networks, Location Management, Geographic Routing, Performance Study

## 1 Introduction

A mobile ad hoc network is a network without preexistent infrastructure. Each node takes on the role of a router for packets not destined for itself. Due to mobility, the network topology varies frequently, and end-to-end sessions are subject to link failures constantly. Majority of current research on ad hoc networks has focussed on designing a routing protocol that scales well with increasing mobility and network size. A performance analysis of popular unicast routing protocols can be found in [1] and [2].

As wireless access become ubiquitous, location based services with user mobility will soon be the norm than the exception, and user location will play a key role in optimizing services for wireless customers. While such services will drive the need for better localization techniques, applications that use locations for optimal performance will require efficient

location management algorithms, where user locations are kept track with minimal system overhead. Geographic routing [3] is one such application that has been suggested for ad hoc networks which make use of the approximate location of nodes for routing data packets, where nodes are aware of their location via the use of a GPS receiver or other localization techniques [4], [5]. Geographic routing lends itself as an attractive candidate for routing in large networks, since it does not use pre-computed routes for packet forwarding during a communication session between two nodes. Such protocols are localized and the amount of state information that needs to be stored by nodes is minimal, and link breakage in a route usually does not affect the end-to-end session. However, the source requires the destination node's location before it can carry out geographic routing.

Thus, location management forms an essential entity in protocols that use geographic routing, in which nodes periodically update select nodes that take on the role of a *location server* of their current location. Location servers can then be queried by source nodes in an on demand fashion for locating destination nodes. Location management usually consists of three phases: location update(choosing location servers and updating them with location information periodically), location maintenance(periodic cleansing of location database for database consistency) and location discovery(querying location servers to obtain a destination's location), and the cost of location management algorithms basically depends upon the cost of the above three primitives.

Many location management protocols have been proposed in literature. Examples of these include the Grid Location Service (GLS)[6], Scalable Location Update based Routing Protocol (SLURP) [7], Scalable Location Management (SLALoM) [8], Uniform Quorum Systems [9], and Hierarchical Grid Location Management (HGRID)[10]. Except [9], all other protocols are grid based protocols, in the sense that, they divide the terrain into an ordered grid, and nodes carry out the location management primitives based on the particular grid ordering of the location management scheme. An upper bound for the average cost of location management has been derived for SLURP, SLALoM and HGRID. In a previous work [11], we analyzed the performance of SLURP, SLALoM and HGRID, for protocol scalability with increasing network size. Our

motivation in this work is two pronged – to study in detail the effect of mobility and traffic load on location management, and to analyze how an increase in these parameters can affect the performance of geographic routing. In this paper, we present our results from the extensive simulations that we carried out to compare the aforementioned protocols. In our simulations, we use a simple geographic routing scheme called *Greedy Forwarding* in which an intermediate node picks its closest neighbor to the destination than itself as the next hop node for packet routing.

The rest of this paper is organized as follows: Section 2 briefly describes the location management schemes under study. In Section 3, we describe our simulation environment and outline the parameters for each protocol. Section 4 describes our comments on the results obtained via simulations and conclude this work in Section 5.

## 2 Location Management Protocols

All the protocols that we describe in this section are grid based: in other words, given a rectangular region of area  $A$ , all protocols divide the topography into  $G$  logical unit regions (also known as *Order-1* regions) where each node is aware of the size of the terrain as well as the size of an unit region. However, they differ from each other in how the unit regions are aggregated for use in location management.

### 2.1 Scalable Update Based Routing Protocol (SLURP)

In SLURP, each mobile node selects exactly one unit region as its *home region* by using a mapping function  $f$ , which uniquely (and randomly) maps its address to the selected home region. The mapping function allows any node to discover another node’s home region simply by knowing its address.

#### 2.1.1 Location Management in SLURP

All mobile nodes which are present in the home region  $R_u$  of node  $u$  act as location servers for  $u$  and keep an entry for  $u$  in their location database. When  $u$  moves across two unit regions  $R_i$  and  $R_j$ , it does the following:

- $u$  updates its home region  $R_u$  of the movement by a location update.
- $u$  also requests nodes in  $R_j$  about the location information it has to keep for nodes that had selected  $R_j$  as a home region.

A home region is updated by sending a location update packet to the region, and the first location server to obtain the packet carries out a broadcast in the region to update all location servers in that region of the movement of  $u$ .

#### 2.1.2 Discovering a node’s location

A source node  $v$  that wishes to communicate with  $u$  can now use  $f$  to identify  $R_u$ , and sends a location query packet towards

$R_u$  to obtain the current location of  $u$ . The first location server to receive the query for  $u$  responds with the current location of  $u$ . As soon as  $v$  receives this response, data packets are routed to this approximate location using the geographic routing algorithm.

## 2.2 Scalable Location Management (SLALoM)

SLALoM combines  $K^2$  Order-1 regions to form Order-2 regions. Each node selects a *home region* in each Order-2 region via  $f$  that maps roughly the same number of nodes to each Order-1 region in an Order-2 region. Hence, every node has  $O(\frac{A}{K^2})$  home regions in  $A$  (note that since the original square cannot be perfectly tiled with Order-2 regions, it is possible that some nodes may not have home regions in the Order-2 regions adjacent to the boundary of  $A$ ). Also, if a node  $u$  is present in an Order-1 region  $R_i$ , which lies in an Order-2 region  $Q_i$ , then all home regions of  $u$  that lie in or adjacent to  $Q_i$  are considered *near* home regions, while the rest are considered *far* home regions.

### 2.2.1 Location Management in SLALoM

When  $u$  moves across two Order-1 regions  $R_i$  and  $R_j$ , it does the following:

- If  $R_i$  and  $R_j$  are in the same Order-2 region  $Q_i$ ,  $u$  informs all its near home regions of the movement, by a partial location update.
- If  $R_i$  is in  $Q_i$ , and  $R_j$  is in a different Order-2 region  $Q_j$ ,  $u$  updates all home regions of the movement by a full location update.
- $u$  also requests nodes in  $R_j$  about the location information it has to keep for nodes that had selected  $R_j$  as a home region.

A single home region update in SLALoM is similar to 2.1.1. Multiple home regions are informed by location updates that traverse a *minimum spanning* tree such that each update traverses a distance  $K$  between two home regions. The length of such a tree is then  $O(\frac{A}{K})$ . Thus it is relatively easy to understand that all home regions know that  $u$  is in  $Q_j$ . In addition, all near home regions know that  $u$  is in  $R_j$ .

#### 2.2.2 Discovering a node’s location

A node  $v$  wishing to communicate to another node  $u$  sends a query packet to the closest home region of  $u$ . If the home region is a near home region, a response is generated, by the location server that gets the query, with  $u$ ’s exact location. If the home region is a far home region,  $v$  forwards its message to the closest near home region of  $u$ , and the location server that receives this message then forwards it to the exact location of  $u$ .

## 2.3 Hierarchical Grid Location Management (HGRID)

For the purpose of clarity, we will refer to Order-1 regions as Level-zero ( $L_0$ ) regions, which form the lowest set of regions in the hierarchy. By combining  $L_0$  regions in groups of 4 and selecting exactly one of those regions in each group, we obtain the next set of leader regions, namely Level-One ( $L_1$ ). A similar procedure of grouping and dividing is done until the hierarchy of  $k$  levels can be established.

The grid hierarchy 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.

### 2.3.1 Location Management in HGRID

When  $u$  moves across two Order-1 regions  $R_i$  and  $R_j$ , the following take place:

- If the movement is within the region under the same  $L_1$  leader,  $u$  sends a location update to  $L_1$ . Otherwise,  $u$  additionally sends an update to its previous  $L_1$  grid indicating its departure.
- At each level  $i$ , the update from  $u$  is processed by the location server that first receives the update, and broadcasts the message in the grid. Further, if the movement specified in the update requires the next hierarchical server to be notified, it forwards the packet to its leader.
- When an 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.

### 2.3.2 Discovering a node's location

To discover a node  $u$ 's location,  $v$  sends a query packet to  $u$ 's  $L_1$  leader. If  $u$  and  $v$  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  $u$ .

## 3 Simulation Environment

We implemented all the three protocols in Glomosim [12] as separate location management layers that operate in conjunction with IP. Data from transport is queued in a separate buffer if the location of the destination is unknown. Packet lifetime in the buffer is 4 seconds, and is subsequently dropped if a location query sent out for the packet's destination fails to return the location of the destination within this lifetime. Apart from having a location database, all 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 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. A periodic broadcast protocol enables each node to realize its local connectivity, and records it in a neighbor table to assist in geographic routing. Greedy forwarding was implemented as the geographic routing algorithm.

Motivated to study the performance of the protocols for robustness with node mobility as well as increasing traffic load, we carried out two sets of simulations. For the first study, we subjected the network to high traffic, while increasing the average node speed. For the second study, the average node speed was kept constant, while varying the traffic conditions in the network. All studies were carried out in a square topography of  $2000m \times 2000m$  consisting of 320 nodes. Specific parameters for our simulations are listed in table I.

TABLE I  
SIMULATION PARAMETERS

Parameters	Scenario I	Scenario II
Simulation Time	300 sec	300 sec
Unit Region Size	250m	250m
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	0-25 m/sec	10 m/sec
Minimum Speed	0 m/sec	0 m/sec
Pause Time	0 sec	0 sec
CBR connections	160	30-210
CBR Rate	2 packets/sec	2 packets/sec
Data Payload	512 bytes	512 bytes
Traffic Pattern	Random	Random
Buffer Size	1000 packets	1000 packets

Since the performance of SLALoM is dependant on the selection of  $K$ , the recommended value of  $K$  from [8] is 7 for our scenario. However, that analysis does not consider the optimal node density for geographic routing, and keeping this in mind as well as the tractability of simulations, we decided to simulate two versions of SLALoM, with  $K = 2$  and  $K = 4$  (hereafter called SLALoM- $K_2$  and SLALoM- $K_4$  respectively). For the scenario considered HGRID defines 3 hierarchical levels. The different packet types and location management overhead in bytes for all protocols are given in table II.

To test the efficiency of the protocols with increase in mobility, we imposed high traffic on the network with 50% of the nodes initiating constant bit rate sessions to other nodes. For the second study, the maximum speed of node was kept constant at  $10m/sec$ , while increasing the number of random

TABLE II  
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

CBR pairs in the network from 30 to 210. Each session has a rate of 2 packets/second, randomly starting after 20 seconds into the simulation and terminating randomly at 250 seconds into the simulation.

## 4 Simulation Results

### 4.1 Results with Increasing 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 1 and 2 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 degree. Since control overhead is least for HGRID (see fig. 5), 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 (as indicated by figure 5). 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 3 and 4.

Figure 3 shows the probability that a query for a destination is returned successfully, and figure 4 shows the average delay

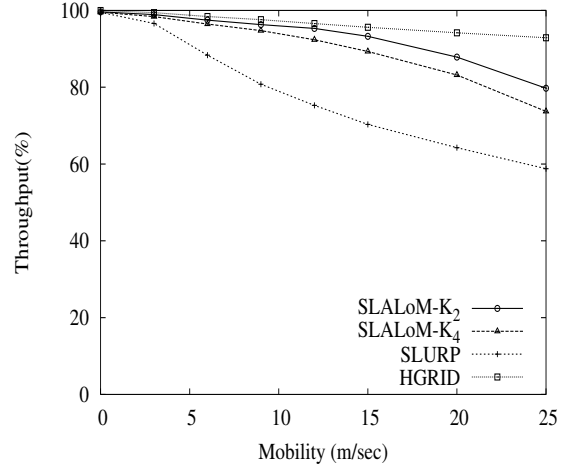


Fig. 1. Data Throughput

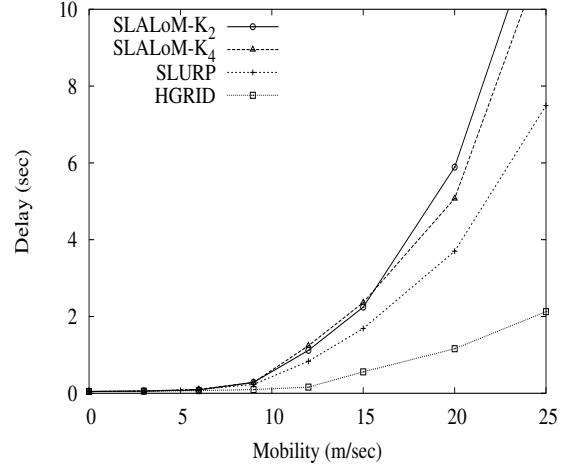


Fig. 2. Average data delay

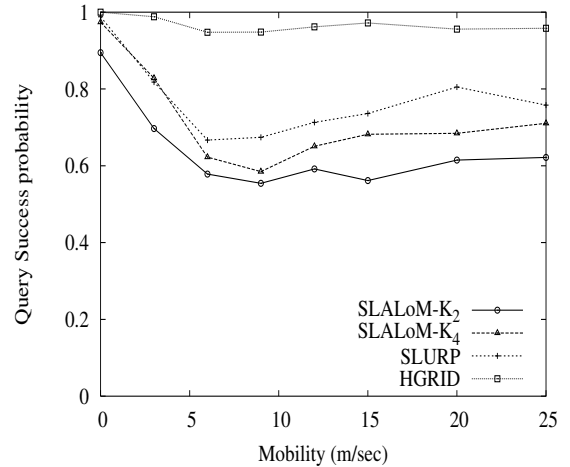


Fig. 3. Query success probability

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.

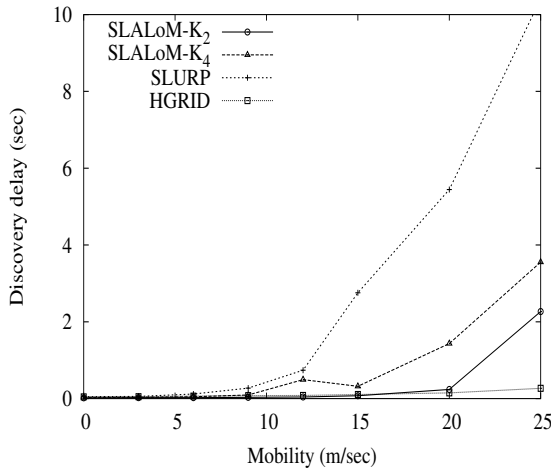


Fig. 4. Average location discovery delay

Figures 5 shows the average control overhead per node incurred by each protocol in number of packets per unit time. 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.

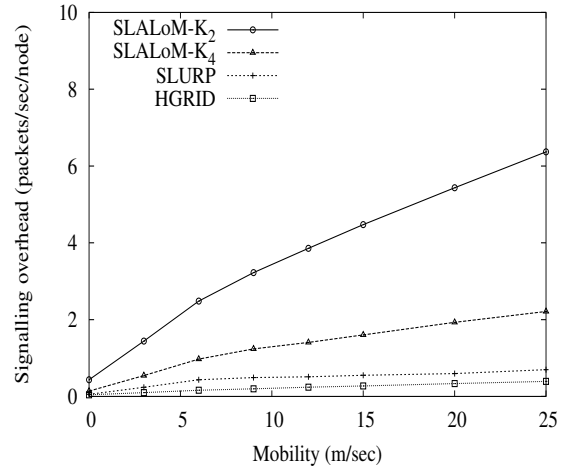


Fig. 5. Control Overhead (packets/sec/node)

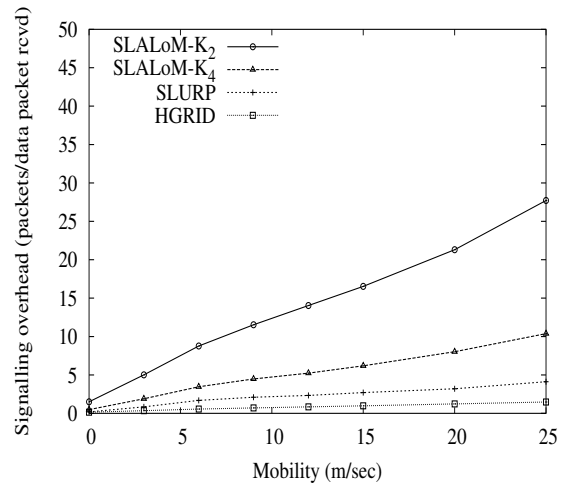


Fig. 6. Control overhead (packets/received data)

## 4.2 Results with Increasing Traffic Load

Figures 7 and 8 show the achieved throughput and delay for data packets. With an average velocity of  $5m/sec$ , the average control overhead is low enough for all protocols to perform nearly equally, with over 90% of the packets being delivered successfully. The average delay increases proportionally to the average control overhead for HGRID and SLURP, with HGRID having a lower average end-to-end delay. Packets have a slightly greater delay in SLALoM- $K_4$  than SLALoM- $K_2$ , although SLALoM- $K_4$  has a lower control overhead. Since update packets are transmitted over greater number of hops in SLALoM- $K_4$ , data packets are often queued up while the control packets are being serviced, resulting in a greater delay.

Finally, figures 9 and 10 show the per node average control overhead for each protocol in packets as well as bytes per received data packet. Although HGRID results in least number of control transmissions, the average number of bytes that the protocol transmits per received data packet is slightly greater than that of SLURP. This can be explained by the size of

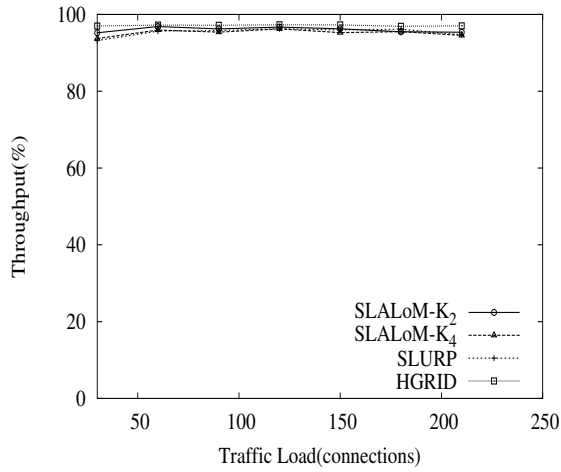


Fig. 7. Throughput

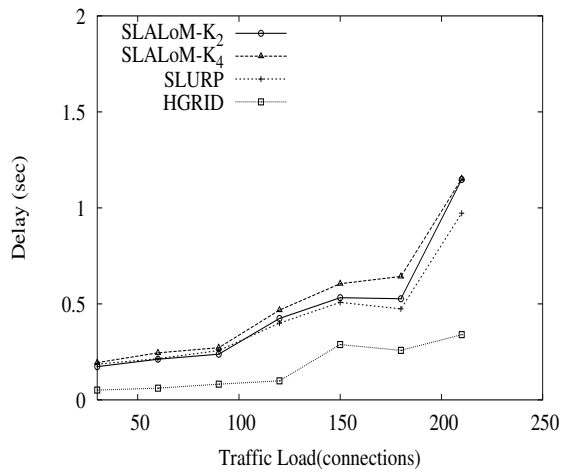


Fig. 8. Average data delay

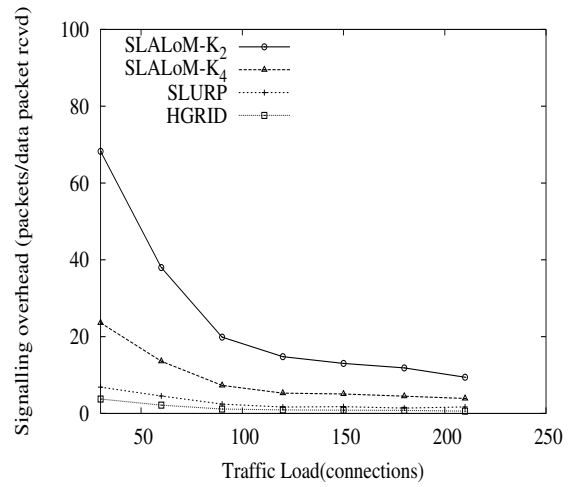


Fig. 9. Control overhead (packets/received data)

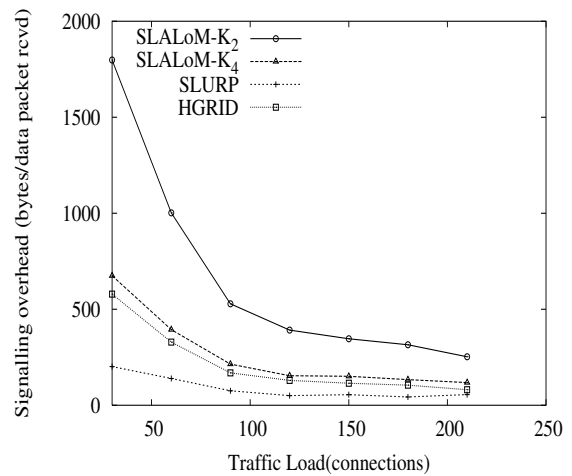


Fig. 10. Control overhead (bytes/received data)

maintenance packets that are exchanged between server nodes in the protocol. 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. Since higher order servers keep track of greater number of nodes, the size of maintenance packets that are exchanged between higher order servers are also greater. Overall, considering the performance with increasing mobility as well as traffic load, HGRID performs much better than SLURP and SLALOM.

## 5 Conclusion

As wireless devices become more capable, location will play a key role in the services offered to customers. While this necessitates inexpensive solutions for efficient location tracking, managing these locations with minimal overhead in systems that use location for optimizing services will become equally important. In this work, we have implemented three grid based location management solutions and carried out

extensive solutions to analyze the efficiency of each and their effect on the performance of geographic routing with by subjecting the network to increasing mobility and traffic conditions. Our results indicate that location management imposes considerable overhead on geographic routing, and can affect the efficiency of the network since control and data share the same channel. All protocols perform well when the network is under loaded and the average mobility of nodes is low. However, the protocols start under performing in a congested network caused by the added overhead due to high mobility. Under low mobility conditions, we noticed that all protocols perform rather well even with high traffic conditions. Thus, an efficient location management protocol is one that resists from congesting the network with increased node mobility, thereby maximizing the bandwidth for data traffic. We also note that a key consideration for designing an efficient location management protocol is the recruitment of location servers such that there is an efficient balance between the update and paging process, leading to reduced control overhead and quick discovery of nodes. In particular, the

Hierarchical Grid Location Management protocol (HGRID) achieved steady through and end-to-end delay for data for our simulations scenarios indicating that it is an efficient contender for location management to assist geographic routing in large mobile ad hoc networks.

## 6 References

- [1] J. Broch, D. Maltz, D. Johnson, Yih-Chun Hu, and J. Jetcheva, "A Performance comparison of multi-hop wireless Ad-Hoc network routing protocols," *Proceedings ACM/IEEE MobiCom*, pp. 85–97, October 1998.
- [2] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance evaluation of two location based routing protocols," *IEEE Infocom*, pp. 1678–1687, 2002.
- [3] B. Karp and H. T. Kung, "GPSR : Greedy perimeter stateless routing for wireless networks," *Proceedings ACM/IEEE MobiCom*, pp. 243–254, August 2000.
- [4] L. Doherty, L. E. Ghaoui, and K. Pister, "Convex Position Estimation in Wireless Sensor Networks," *IEEE Infocom*, April 2001.
- [5] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," *IEEE Infocom*, 2003.
- [6] 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.
- [7] S. Woo and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 7, pp. 513–529, January 2001.
- [8] 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 IEEE Wireless Communications and Networking Conference*, March 2002.
- [9] 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.
- [10] S. J. Philip and C. Qiao, "Hierarchical grid location management for large wireless ad hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review(MC2R)*, vol. 7, no. 3, pp. 33–34, 2003.
- [11] S. Philip, J. Ghosh, S. Khedekar, and C. Qiao, "Scalability analysis of location management protocols for mobile ad hoc networks," *Proceedings of the IEEE WCNC*, 2004.
- [12] 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.