# Routing on Overlay Graphs in Mobile Ad Hoc Networks

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

Joy Ghosh, Hung Q. Ngo, Chunming Qiao
Department of Computer Science and Engineering
SUNY at Buffalo
Amherst NY 14260
Email: {joyghosh,hungngo,qiao}@cse.buffalo.edu

***Abstract*— Geometric routing using source-destination locations has been widely suggested as a scalable alternative to conventional routing approaches in mobile ad hoc networks. Recently, there has been considerable attention on face routing in planar graphs constructed from overlay graphs in wireless networks. Given a plane tiled into an infinite mesh of polygons, an overlay graph is defined as one in which a graph edge is defined between two adjacent polygons if a radio link exists between any two nodes located in these polygons. We consider the problem of constructing a connected planar graph from the overlay graph, and geometric routing in such graphs. We prove a specific property of such graphs known as the *redundancy property* and propose a distributed routing algorithm called Grid Traversal Algorithm (GTA) based on the redundancy property of overlay graphs. The algorithm is both localized and energy efficient, but may not guarantee connectivity in pathological cases. Simulations show that such disconnections are rare in practise and that GTA performs very well in terms of percentage of data delivered, data delay and overhead compared to GPSR, a geometric routing protocol that routes on a planar graph extracted from the unit disk graph.**

## I. Introduction

Greedy geometric routing (also known as position/location based or geographic routing) has been widely suggested as an alternative to conventional routing approaches in mobile ad hoc networks in lieu of routing scalability. In geometric routing, it is assumed that mobile nodes are aware of their own location via the use of a GPS receiver or other localization schemes. A localized periodic broadcast protocol enables all nodes to have approximate knowledge of their neighbors' locations, and a location management protocol [1] can be used to discover the position of the destination node. Several proposals have been described in literature that make use of a node's neighborhood position knowledge and greedy forwarding to route packets to a known location of the destination node [2], [3], [4].

However, it is well known that greedy routing suffers from the local maxima problem, in which greedy forwarding fails due to a void in the path. In such cases, packets are dropped at the intermediate node even if a perfectly valid path exists between the source node and the destination node. To recover from the error caused by greedy forwarding, face routing on a planar graph extracted from the static wireless network was considered in [5], [6]. The wireless network is modeled as a geometric *unit disk graph* (UDG), in which the Euclidean co-ordinates of the mobile node represents a vertex in the plane, and an edge exists between two vertices if the Euclidean distance between the vertices is less than the node transmission range $R$. In general, the UDG is not planar, but known techniques such as [7] can be used to extract a sub-graph from a connected UDG which is both planar and connected. Face routing on the planar graph then guarantees the delivery of packets in a static wireless network.

Recently, there has been some attention to routing in planar graphs constructed from an *overlay graph* of the wireless network [8]. The overlay graph (referred to as *geographic clusters* in [8]) is defined as follows: the plane is partitioned by a regular infinite mesh of polygons (e.g. hexagons), each forming a geographic cluster. Each node is assigned to exactly one cluster by its current location. Two clusters $C_1$ and $C_2$ are adjacent if there are two connected nodes, one lying in $C_1$, and the other in $C_2$. The diameter of each cluster is the transmission range of a node, and the graph $G$ defined by adjacent clusters is termed the overlay graph. To make $G$ planar, a localized Gabriel Graph construction is investigated, and the result of the study is that a planar overlay graph can be constructed from one–hop neighbor information, but may produce a disconnected subgraph even if the unit disk graph is connected.

In this paper, we introduce a distributed algorithm to construct a planar overlay subgraph by using a specific property of overlay graphs known as *redundancy property* and by adding *virtual* edges to the original graph. While the subgraph constructed by the distributed algorithm is always planar, there are pathological cases where the algorithm cannot guarantee connectivity. We investigate a routing protocol that performs geometric routing on a planar overlay subgraph (known as GTA) and show by simulations that such disconnections occur rarely in practise. Further, we show that routing on a planar overlay subgraph is superior to GPSR, a geometric routing protocol that routes on a planar subgraph constructed from the UDG.

The rest of this paper is organized as follows: Section II describes an overlay graph and the connected overlay planar graph construction problem. We propose a distributed algorithm to construct such a subgraph in Section III and

1-4244-0357-X/06/$20.00 ©2006 IEEE
*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*

introduce the GTA routing protocol in section IV. We present the results from simulations that show the effectiveness of our solution in Section V and conclude this work in Section VI.

## II. Connected Overlay Planar Graph Construction Problem

Given a set of mobile nodes $N$ and their Euclidean co-ordinates, the network terrain is tiled into an uniform infinite grid such that the unit square regions within the grid are of diameter $r$, where $r$ is the radio range of a node. The center point $(x_i, y_i)$ of each unit region $R_i$ uniquely represents the unit region and forms a vertex in the overlay graph. A total ordering on the vertices are defined as follows:

- $(x_1, y_1) < (x_2, y_2)$ if $x_1 < x_2$ or $x_1 = x_2$ and $y_1 < y_2$
- $R_i < R_j$ if $(x_i, y_i) < (x_j, y_j)$

A mobile node $u \in N$ can now be uniquely assigned to one of the vertices by defining a total ordering on the node locations on the grid using the following three rules:

- A node $u(x_u, y_u)$ is assigned to $R_i$ if $(x_u, y_u) \in R_i$
- A node $u(x_u, y_u)$ is assigned to min$(R_i, R_j)$ if $(x_u, y_u) \in R_i, R_j$
- A node $u(x_u, y_u)$ is assigned to min$(R_i, R_j, R_k, R_l)$ if $(x_u, y_u) \in R_i, R_j, R_k, R_l$

The last two rules assign nodes to a unique vertex if they fall on the boundary of two unit regions or the corner of four unit regions.

Once the vertices are defined, the adjacency matrix of the overlay graph $G(V, E), R_i \in V$ is defined as follows: two vertices $R_i$ and $R_j$ are adjacent, and a bidirectional edge $e_{R_i \rightarrow R_j} \in E$ exists between them if there is a node pair $u, v$ $(u, v \in N, u \in R_i, v \in R_j)$ that are directly connected to each other in the UDG. Due to this construction, a vertex $R_i$ can be adjacent to at most 20 other vertices as shown in figure 1.

To facilitate the explanation of our distributed algorithm described later, we denote by $E_s^{R_i}$ the set of eight short edges (shown in bold solid lines in Figure 1) connecting the eight immediate vertices around $R_i$, and by $E_l^{R_i}$ the twelve remaining edges incident from $R_i$ (shown in dashed lines in Figure 1). An edge is also classified as either an *axis*-edge or a *non-axis*-edge depending on whether it is parallel to the $X - Y$ (horizontal/vertical) axes or not. Denote by $N_i$ the neighborhood of $R_i$, which consists of the 20 vertices to which $R_i$ may be adjacent. With the overlay graph in place, the connected overlay planar graph construction problem is defined as follows: construct a subgraph $G_P(V, E_P), E_P \subseteq E$ which is both planar and connected.

In [8], Frey et. al. studied a similar problem in which the plane is tiled into a mesh of hexagons and each hexagon is referred to as a geographic cluster. Nodes are aggregated into geographic clusters based on their position in the plane, and an aggregate graph is constructed from the radio connectivity between adjacent hexagons. A planar subgraph embedded in the aggregate graph is extracted by a modified Gabriel graph [9] construction using one-hop information advertised by each node in a beacon packet. Routing is then performed on the planar subgraph extracted from the aggregate graph.

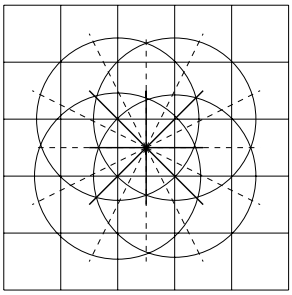

Fig. 1. A vertex in the overlay graph is adjacent to at most 20 other vertices. Short edges are shown by the bold solid lines while dashed lines indicate long edges. While axis edges are parallel to the grid , non–axis edges are inclined to the grid.

*Lemma 1: The Gabriel graph construction proposed in [8], when applied to the overlay graph, will always produce a planar subgraph, but the resulting subgraph may be disconnected even if the underlying unit disk graph is connected.*

*Proof:* The proof follows directly from [8]. As an example, consider the node placements in figure 2. Nodes $v_1, v_2, v_3$ are assigned to vertices $R_j, R_i, R_k$ respectively. Since $v_1$ and $v_3$ are disconnected, there are two bidirectional edges $e_{R_i \rightarrow R_j}$ and $e_{R_i \rightarrow R_k}$ in the overlay graph, indicated by the dashed lines. In the Gabriel graph construction for a node pair $(u, v)$, there must be no witness node located in the disk with diameter $|uv|$ other than $u$ and $v$. Thus, to preserve planarity in the overlay graph consisting of vertices $R_i, R_j$ and $R_k$, edge $e_{R_i \rightarrow R_j}$ will be discarded, thereby leaving the subgraph disconnected. ■

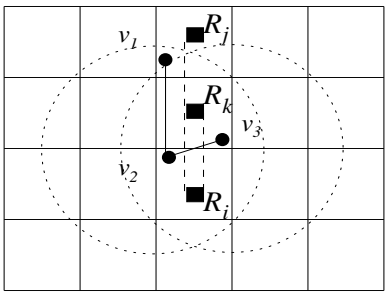


Fig. 2. The circles indicate the radio range of nodes $v_2$ and $v_3$. Dots indicate nodes, solid lines indicate edges in the unit disk graph, solid squares represent the graph vertices and the dashed lines indicate edges in the overlay graph.

From lemma 1, we note that it may not be possible to construct a planar subgraph that only has a subset of the original edges of the overlay graph by the process of edge removal. As a solution, we slightly modify the definition of overlay graphs to include *virtual edges* to the set of overlay edges to preserve planarity as well as connectivity. A virtual edge is simply a short edge between adjacent vertices (not originally in the overlay graph) which are realized via 2-hop radio paths in the underlying unit disk graph. An algorithm that adds virtual edges to the overlay graph will be presented in section III. Next, we present a specific property of overlay graphs known as the *redundancy property*, which can be used to eliminate

1-4244-0357-X/06/$20.00 ©2006 IEEE
*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*

edge intersections to create a planar overlay subgraph.

*Lemma 2:* **Redundancy Property***: If two edges $e_1$ and $e_2$ in the overlay graph intersect, then amongst the two mobile node pairs $(u, v)$ and $(x, y)$ that constituted the overlay graph edges, there must be at least one node which is directly connected to the remaining three nodes in the unit disk graph.*

*Proof:* The proof is omitted due to space limitation, and can be found in [10]. ■

Due to lemma 2, if two edges in the overlay graph intersect, then one of the intersecting edges is redundant since there must be two additional overlay graph edges that connect all the four nodes under consideration. Thus, one of the intersecting edges can be removed without disconnecting the node pairs. An algorithm that uses the redundancy property for removing redundant edges to create a planar subgraph from the overlay graph will be described in section III.

## III. A Distributed Algorithm for Overlay Planar Graph Construction

We now describe a distributed algorithm that constructs a planar subgraph from the overlay graph by adding virtual edges and by removing redundant edges using the redundancy property. The algorithm consists of two phases: the first phase creates a set of edges from an adjacency table that could potentially be part of the final edge set of the subgraph, and the second phase removes intersecting edges from this set to preserve planarity. To realize 1-hop overlay edge connectivity and to facilitate geometric routing, each node in region $R_i$ periodically broadcasts its location, and updates an adjacency table if it either hears a broadcast from a neighbor node in an adjacent region $R_j$, or a broadcast from a neighbor node in $R_i$ which advertises an overlay edge to adjacent region $R_j$. Each node then includes its incident overlay graph edge information in the subsequent periodic broadcasts. Note that this information can be efficiently coded in a bitmap consisting of 20 additional bits in the periodic broadcast, since there are at most 20 vertices that any vertex can be adjacent to in the overlay graph. After gathering adjacent region connectivity from such beacons, each node runs algorithm 1 (Phase $I$) and algorithm 2 (Phase $II$) to compute the planar graph.

*1) Algorithm Description:* Lines $1-5$ of algorithm 1 add the short edges, and lines $6-25$ add the long edges between any two vertices in the overlay graph to the potential edge set $E_P$. While non–axis long edges can be readily added to the edge set (lines $21-22$), long axis edges have to be added cautiously. Particularly, node configurations such as the ones shown in Figures 3 and 4 can cause overlay edge intersections along infinite points. For example, since all nodes are physically connected, there will be three edges in the overlay graph in figure 3 - namely, two short edges $e_{R_i \to R_k}$, $e_{R_k \to R_j}$ and a long edge $e_{R_i \to R_j}$. Since the long edge intersects with the short edges at infinite points, it can be discarded without loss of connectivity due to the existence of the path $R_i R_k R_j$ (lines $10-11$).

On the other hand, if the intersection due to axis edges is caused by a long edge and a short edge, and if the removal of one of the edges would cause disconnection, the algorithm does the following: it discards the long edge, and adds a virtual edge between the vertices that did not previously have a short edge between them (lines $12-17$). The virtual edge represents a 2-hop physical path in the underlying UDG. For example, in figure 4, long axis edge $e_{R_i \to R_j}$ is discarded and a virtual edge $e_{R_k \to R_j}$ is added to the edge set $E_P$. Edge $e_{R_k \to R_j}$ is constituted by the physical path $v_3 \to v_2 \to v_1$. Finally, If there are no intersections between axis edges, then the long edge can be safely added to $E_P$(lines $18-19$).

**Algorithm 1** To add potential edges to $E_P$

```
1:  for all R_i in adjacency table do
2:    for all e_{R_i→R_j} ∈ E_s^{R_i} do
3:      Add e_{R_i→R_j} to E_P
4:    end for
5:  end for
6:  for all R_i in adjacency tabl do
7:    for all e_{R_i→R_j} ∈ E_l^{R_i} do
8:      if e_{R_i→R_j} is an axis-edge then
9:        R_k ← vertex incident on e_{R_i→R_j}
10:       if e_{R_i→R_k} && e_{R_k→R_j} then
11:         continue
12:       else if e_{R_k→R_j} then
13:         Discard e_{R_i→R_j}
14:         Add virtual edge e_{R_i→R_k} to E_P
15:       else if e_{R_i→R_k} then
16:         Discard e_{R_i→R_j}
17:         Add virtual edge e_{R_k→R_j} to E_P
18:       else
19:         Add e_{R_i→R_j} to E_P
20:       end if
21:     else
22:       Add e_{R_i→R_j} to E_P
23:     end if
24:   end for
25: end for
```

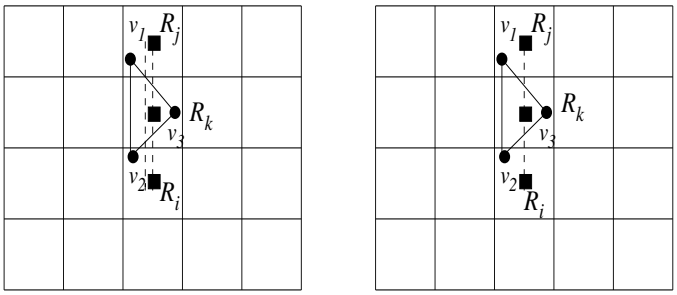


Fig. 3. Long edge $e_{R_i \to R_j}$ can be discarded since the edge vertices $R_i$ and $R_j$ are reachable via the short edges $e_{R_i \to R_k}$ and $e_{R_k \to R_j}$.

Once algorithm 1 completes, edges in $E_P$ can only intersect at a single point. Algorithm 2 then considers all edge pairs that intersect at a single point (other than the vertices themselves) to make the subgraph non–planar. Due to the redundancy

1-4244-0357-X/06/$20.00 ©2006 IEEE
*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*

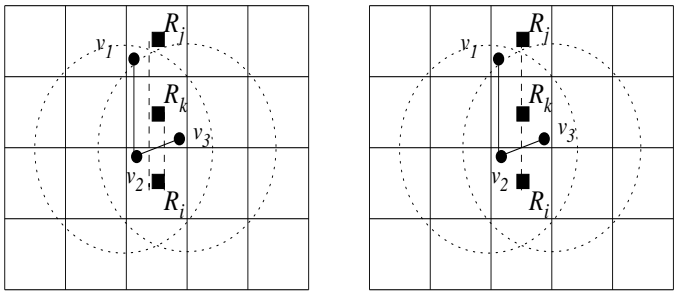


Fig. 4. A short virtual edge $e_{R_k \to R_j}$ is introduced to eliminate the long edge $e_{R_i \to R_j}$. The virtual edge $e_{R_k \to R_j}$ represents the path $v_3 \to v_2 \to v_1$ in the unit disk graph.

**Algorithm 2** To compute planar overlay subgraph

```
1:  for all e_{R_i→R_j} in E_P do
2:      l(e_{R_i→R_j}) ← max(i, j)
3:  end for
4:  for all e_{R_i→R_j} ∈ E_P do
5:      for all R_k ∈ N_i do
6:          for all e_{R_k→R_l} ∈ E_P do
7:              if e_{R_i→R_j} and e_{R_k→R_l} intersect then
8:                  if e_{R_i→R_k} ∈ E_P && e_{R_i→R_l} ∈ E_P then
9:                      if e_{R_k→R_i} ∈ E_P && e_{R_k→R_j} ∈ E_P then
10:                         remove edge with lower label
11:                     else
12:                         remove e_{R_k→R_l} from E_P
13:                     end if
14:                 else
15:                     remove e_{R_i→R_j} from E_P
16:                 end if
17:             end if
18:         end for
19:     end for
20: end for
```

property, any non–planarity in $E_P$ can be independently detected by at least one node in the node pairs that caused the edge intersection. Hence one of the intersecting edges can be removed by this node to preserve planarity. In case of intersecting edge pairs and vertices that are completely connected, the tie is broken by removing the edge having a lower label $l(e_{R_i \to R_j})$ (lines 7–10).

Once each node computes its final edge set in the overlay graph, it includes this information in the periodic beacon using an additional 20 bit flag. A node that receives such a broadcast packet from its neighbor updates its connectivity, and marks the outgoing overlay graph edges to be preserved as advertised in the broadcast beacon. Clearly, this scheme is purely distributed, localized and energy efficient in nature. The overhead required to construct an overlay planar subgraph is at most 40 bits more than that required to keep track of the location of radio neighbors to facilitate geometric routing. However, it is possible that the graph which is locally constructed -albeit planar-, is disconnected, as shown by theorem 3.

*Theorem 3: The subgraph constructed locally by the distributed algorithm is always planar, but does not guarantee connectivity.*

*Proof:* Since both algorithms remove one of the intersecting edges in all intersecting edge pairs, the resulting subgraph is always planar. Lack of connectivity is demonstrated by a conflict graph in which disconnection occurs. In figure 5, nodes $u$,$v$,$w$,$x$ and $y$ can be placed in overlay vertices $R_5$, $R_{10}$,$R_{11}$,$R_2$ and $R_3$ respectively to produce the overlay graph as shown. For the intersection between edges $R_5 \to R_{11}$ and $R_2 \to R_{10}$, edge $R_2 \to R_{10}$ will be removed since all the four vertices in question can be reached by either vertices $R_{10}$ or $R_{11}$, and the tie is resolved by removing the lower labeled edge; i.e., $R_2 \to R_{10}$. For the intersection between edges $R_3 \to R_{10}$ and $R_2 \to R_{11}$, edge $R_2 \to R_{11}$ will be removed since vertex $R_{11}$ cannot connect to vertex $R_3$ directly. As a consequence, vertex $R_2$ will be disconnected in the final subgraph. ■

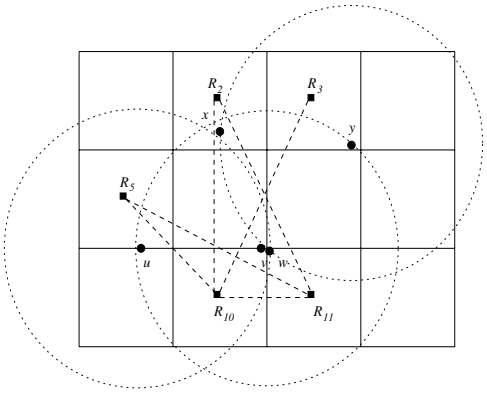


Fig. 5. Proof of theorem 3

Although the distributed algorithm cannot guarantee connectivity by the redundancy property alone, this may not be a severe problem since the cases in which specific node placements that lead to disconnection may occur rarely in practice. In section V, we will show that this is indeed the case in most practical scenarios.

## IV. The GTA Routing Protocol

Using the distributed algorithm explained in section III, we introduce a new routing protocol that we call the Grid Traversal Algorithm (GTA). Routing in GTA is ordinarily greedy forwarding using the neighbor location table. However, the routing switches to face routing mode when greedy forwarding fails due to the occurrence of local maxima, similar to face routing schemes such as GFG [5] and GPSR [6]. The key difference in GTA is that face routing is carried out on a planar graph constructed from the overlay graph rather than by the one extracted from the unit disk graph. Face routing guarantees that the exploration of the faces that lie between the occurrence of local maxima and the destination node will eventually lead to the delivery of the packet to the destination if the network is static and connected. On the other hand, if

1-4244-0357-X/06/$20.00 ©2006 IEEE
*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*

the network is disconnected, the packet will loop back to the unit region $R_i$ where face routing mode was initiated. Since this information is recorded in the packet, the first node in $R_i$ to receive the packet that loops back to $R_i$ will drop the packet.

## V. Numerical Study

### A. Simulation Scenario

We compared the performance of GTA against GPSR by implementing both protocols in GloMoSim [11]. GTA has an adjacency table to keep track of adjacent vertices that are reachable, and a list of next best hops to reach each adjacent vertex. A boolean variable is associated with each graph edge to indicate its membership in the planar graph. Both protocols also have a neighbor table that keeps track of 1-hop neighbor nodes' position information to facilitate geometric routing. GPSR implementation was borrowed from the NS–2 implementation of the protocol at [12] and verified against published results by running the protocol using an ideal location management layer. The location of a destination node is known *a priori* using the *ideal* location management scheme. The RNG scheme [7] was used to create the planar graph for perimeter routing in GPSR, since this scheme yields a less densely connected graph and leads to better performance of the routing protocol.

Parameters for the simulation scenario are shown in Table I. For the simulation scenario, 1000 Constant Bit Rate (CBR) connections were randomly generated, with each session sending one packet with a 512 byte data payload. A session terminates successfully if the data packet sent to the destination's location successfully reaches the destination before the simulation ends. Nodes move according to the Orbit mobility model [13], and in this scenario, nodes move on a rectangular road that connects four rectangular hubs located at four corners of the simulation terrain. As soon as a node reaches a hub, a random point in the next hub is selected and a random velocity between $[V_{min}, V_{max}]$ is selected to move to the next hub. This mobility model was intentionally chosen so that face routing dominated over greedy forwarding for packet routing.

### B. Simulation Results

Each plot point presented in this section is an average of five simulation runs. Figure 6 shows the fraction of data packets delivered as a function of maximum node speed. From the figure, it is clear that GTA is able to perform steadily, delivering nearly 100% of all data packets while GPSR is clearly not able to deliver all data packets. While face routing guarantees packet delivery when the network is static, this is no longer true when the graph faces change due to node mobility. Face routing in planar overlay graphs is clearly advantageous since the overlay graph faces remain relatively stable with increasing node mobility than those in a planar graph constructed from the UDG. As long as there is connectivity between two unit regions, there is an edge in the overlay graph, irrespective of the *actual* nodes that are involved in the connection. On the other hand, changes in UDG due to node mobility directly impacts the UDG planar graph constructed.

Figure 7 shows the average end–to–end packet delay for successfully delivered packets using both GTA and GPSR. Node density has a direct impact on the planar graph constructed from the UDG for GPSR. As the number of nodes increases, longer edges are replaced by multiple shorter edges, leading to longer paths and delays for GPSR. As for GTA, node density does not impact the planar graph constructed to the same extent, and path lengths remain more or less the same irrespective of the number of nodes. In fact, increasing node density is helpful for GTA since it increases connectivity between adjacent unit regions. Clearly, 250 nodes moving along the edges of the terrain represents high node density, and this adversely affects GPSR path lengths.

Finally, Figure 8 shows the number of data transmissions required to deliver the data packets for both the protocols. There is a significant increase for this metric with GPSR when compared with GTA. Clearly, one reason is due to the longer paths in GPSR which causes additional transmissions that lead to an overall increase in the number of transmissions. Secondly, as node mobility increases, graph faces change and the destination may no longer be reachable via face routing. In such cases, packets loop around graph faces before they are dropped at the node where face routing is initiated. In GPSR, such loops may be long and packets may float around for long periods before IP drops them due to the TTL limit. Such problems can be avoided using limited graph edges in planar overlay graphs as constructed by GTA. Thus, GTA emerges as a clear winner in terms of packet delivery, average end–to–end delay and data overhead in comparison with GPSR. It is noteworthy that the disconnection problem does not adversely affect GTA in average cases such as the ones simulated in our study.

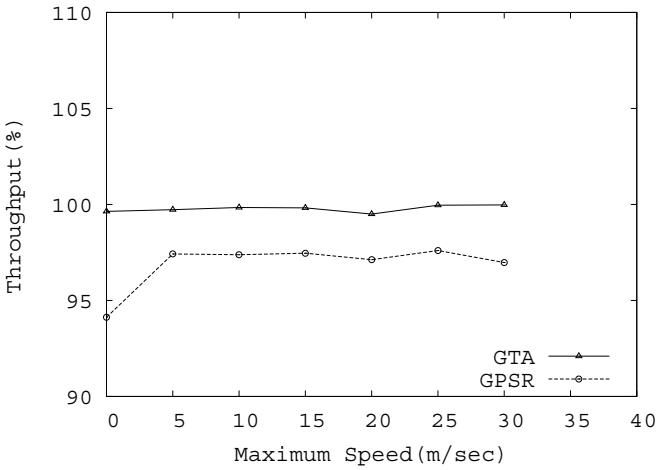

Fig. 6. Percentage of data delivered

## VI. Conclusion

Geometric routing takes advantage of location information to facilitate routing, and is widely believed to be scalable for

1-4244-0357-X/06/$20.00 ©2006 IEEE
*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*

TABLE I

SIMULATION PARAMETERS

| | | | |
|---|---|---|---|
| Simulation Time | 1000 sec | Mobility Model | Orbit Mobility |
| Simulation Area | 2000×2000m | Maximum Speed | 0-30 m/sec |
| Unit Region Size | 250m | Minimum Speed | 1 m/sec |
| Number of Nodes | 250 | Pause Time | 0 sec |
| Transmission Range | 350m | Traffic Type | Random CBR |
| Transmission Speed | 54 Mbps | Number of Connections | 1000 |
| MAC Protocol | IEEE 802.11g | Data Payload | 512 bytes |
| Beacon Interval | 1 sec | Buffer Size | 1000 packets |

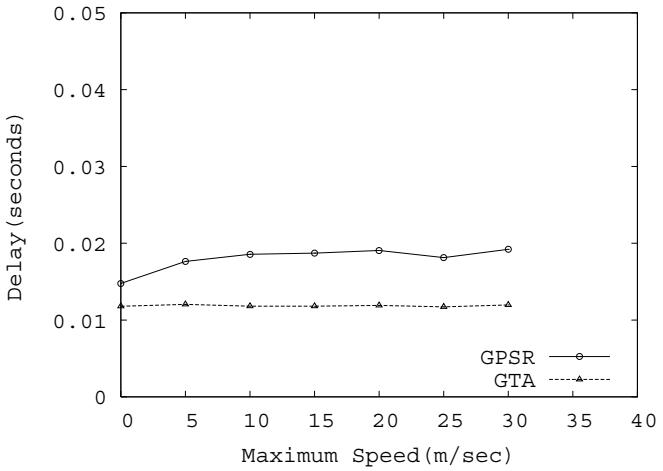


Fig. 7. Average data delay

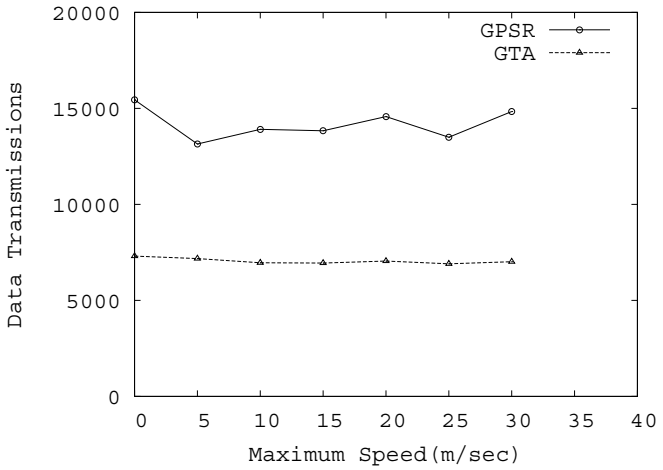


Fig. 8. Total data transmissions

routing in ad hoc networks when compared with conventional routing protocols that are table driven or on demand. We have considered the concept of overlay graph routing, and the connected overlay planar graph construction problem. Overlay graphs are potentially useful in many applications, including robust and efficient routing in mobile networks. To our knowledge, there is no other technique that extracts a connected planar subgraph from an overlay graph using edge removal. We have presented a distributed algorithm to construct a planar subgraph from the overlay graph by using the redundancy property of overlay graphs, and by adding virtual edges. We have also presented a geometric routing protocol that routes on the planar overlay subgraph, which is robust in the face of node mobility. While the subgraph constructed by the distributed algorithm is always planar, it cannot guarantee connectivity in pathological cases. Using simulations, we have shown that such disconnections occur rarely in practice and that the protocol performs much better in terms of data delivered, average end–to–end delay and data overhead in comparison to GPSR, a geometric routing protocol that routes on a planar subgraph constructed from the unit disk graph. Our protocol is localized, energy efficient, and highly effective for routing in mobile ad hoc networks.

## REFERENCES


[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] Toussaint G.T, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261–268, 1980.

[8] H. Frey and D. Goergen, "Planar graph routing on geographical clusters," *Ad Hoc Networks*, vol. 3, no. 5, pp. 560–574, 2005.

[9] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, pp. 259–278, 1969.

[10] Sumesh J. Philip, "Scalable Location Management for Geographic Routing in Mobile Ad hoc Networks," *Computer Science Tech Report TR-2005-21, SUNY at Buffalo*, 2005.

[11] 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.

[12] B. Karp, "Greedy perimeter stateless routing (GPSR)," *http://www.icir.org/bkarp/gpsr/gpsr.html*.

[13] Joy Ghosh, Sumesh J. Philip and Chunming Qiao, "Sociological orbit aware location approximation and routing (SOLAR) in MANET," *Ad Hoc Networks*, Article in Press.



1-4244-0357-X/06/$20.00 ©2006 IEEE

*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.*