A Distributed Graph Algorithm for Geometric Routing in Ad Hoc Wireless Networks

Rashid Bin Muhammad
Department of Computer Science
Kent State University, Kent, Ohio, USA
Email: rmuhamma@cs.kent.edu

***Abstract*— This paper presented a fully distributed algorithm to compute a planar subgraph of the underlying wireless connectivity graph. This work considered the idealized unit disk graph model in which nodes are assumed to be connected if, and only if, nodes are within their transmission range. The main contribution of this work is a fully distributed algorithm to extract the connected, planar graph for routing in the wireless networks. The communication cost of the proposed algorithm is $O(d\ log\ d)$ bits, where d is the degree of a node. In addition, this paper also presented a geometric routing algorithm and established its lower bound. The algorithm is fully distributed and nodes know only the position of other nodes and can communicate with neighboring nodes in their transmission range.**

***Index Terms*— Geometric routing, unit disk graph, Delaunay triangulation, Gabriel graph, lower bound.**

I. Introduction

Wireless ad hoc networks consists of independent devices (transceiver nodes) communicating via radio. There is no common infrastructure, which could be used for the organization of the network. A typical ad hoc network is composed of heterogeneous devices. For a successful setup of the communication, it is essential that these diverse devices e.g. cell phone, PDAs, laptops, etc. be able to communicate with each other. Due to these characteristics, wireless ad hoc networks are adopted in many military and civil applications such as battlefield surveillance, environmental control, and security management. In these applications, the connectivity of links and routing in the connected network are important architecture and algorithmic issues, respectively.

In this paper, we consider the problem of routing in ad hoc networking for which hosts know nothing about the network except the locations of the hosts to which they can communicate directly. In particular, we consider the case in which all hosts have the same broadcast range. Our algorithm is a geometric routing algorithm for which we give a constructive lower bound. In the literature algorithms of this nature are known as geographic, location-based, and position based algorithms. Formally, given a Euclidean graph $G=(N,E)$ with $|N|=n$, the goal of our geometric algorithm is to convey a message from a source $s\in N$ to a destination $t\in N$ by sending information packets over the edges of graph G while obeying the following conditions. Firstly, at the start, all nodes $v\in N$ in G know their geometric positions (coordinates) and geometric positions (coordinates) of all of their neighbor-nodes within transmission range R. Note that this assumption becomes more and more realistic with the introduction of inexpensive and miniaturized positioning systems. Secondly, the source node $s\in N$ knows the position i.e. coordinates, of the destination node $t\in N$ in the given graph G. For this assumption, a (peer-to-peer) overlay network could be employed [1], [2]. Lastly, the nodes $v\in N$ are not allowed to store anything except for temporarily storing packets before transmitting them to the neighbor node(s). Furthermore, a packet can store the information not larger than $O(\log n)$ bits, i.e. information about $O(1)$ nodes is allowed. Overviews of geometric routing algorithms are given in [1], [3], [4].

This paper investigates the problem of extracting the connected, planar subgraph from the given graph for geometric routing algorithms in ad hoc wireless networks. The preliminary versions of this work appeared in References [5], [6]. In a typical ad hoc network, most of the nodes are mobile but here we assume that routing takes place much faster than node movement. In other words, mobility is not considered here. In addition, we assume that the routing layer can access the location information about network nodes. We shall formulate this problem in the geometric graph as follows. Let N be a set of nodes deployed in a certain region R, with $|N|=n$. The problem is to build a planar graph $G=(N,E)$ on N such that each node is connected to its closest neighbors. Formally, the edge $(u,v)\in E$ if, and only if, $\delta(u,v)\leq 1$, where $\delta(u,v)$ is the distance between node u and its closest neighbor v.

In addition, we present a distributed algorithm for routing on the unit disk graph. While the fundamental approach is based upon the famous Face Routing algorithm [7] (in the original paper the algorithm is called Compass Routing II), the main difference is that during traversal of G when the algorithm crossed by the line segment joining the source and the destination nodes, it moves to the adjacent face as suggested in [8].

The general outline of the paper is as follows. In the next section we discuss the related work. The Section III presents formal computational model of ad hoc wireless

This paper is based on “Distributed Construction of Planar Graph for Position-Based Routing in Ad Hoc Wireless Networks,” by Rashid Bin Muhammad, which appeared in the Proceedings of the IEEE SouthEast Conference (SoutheastCon 2007), Richmond, VA, USA, March 2007.

networks and some basic definitions presented in Section IV. In Section VI we present the overall strategy of the algorithms. The Section VII presents the fully distributed algorithm for the construction of planar graph and presents the analysis of the algorithm. In Section V we discuss some major reasons, which lead to the failure of planar graph construction. In Section VIII we present and analyze the routing algorithm. The Section IX establishes the lower bound for the routing algorithm on the planar graph. Finally, some concluding remarks are collected in the Section X.

II. Relevant Literature

To the best of our knowledge, Takagi [9] is the first who proposed geometric routing. Specifically, Takagi solved the problem of the maximum expected progress per hop in the network with the condition that the optimal transmission probability of nodes is in some specified range, say N. The paper [9] also talked about the various consequences, especially performance, on a couple of computational models. After this first proposition of geometric routing, we have seen quite a number of works such as [10] and [11] that proposed algorithms based on various notions of progress of information packet or message from source to destination. The main problem with those algorithms was that most of them do not necessarily guarantee message delivery. Other problem, which needs attention, is the congestion. As an example, flooding routing algorithms [12], multiple redundant copies of the messages are sent in the hope that one of them will eventually reach its destination. Clearly, sending multiple copies of messages creates the problem of network congestion. The early algorithms for routing in ad hoc networks used greedy approach [9], [11], [13]. The basic idea of those algorithms was as follows. A current node forwards an information packet to its neighbor node that is closest to the destination node. However, most of those algorithms failed to delivered packet to the destination and fall into a local minimum because of the lack of "better" neighbors.

To the best of our knowledge, the reference [7] presented the first "correct" Face Routing algorithm that guaranteed packet delivery. The main idea of the Face Routing algorithm is that an information packet walks along faces of planar graphs and proceeds along the line connecting the source and destination nodes. Since we borrowed the basic idea from the famous Face Routing algorithm introduced in [7] (in the original paper the algorithm is called Compass Routing II) and for completeness we describe the algorithm here. Start at source s and let F be the face that is intersected by line segment joining source s and destination t, $\overline{st}$. Explore the boundary of face F by traversing the edges of F and remember the intersection point p on line $\overline{st}$ with the edge of F which is nearest to destination t. After traversing all edges, go back to p. if we reach the destination while traversing the boundary of F, we are done. Otherwise, p divides the line $\overline{st}$ into two line segments where line pt is not yet "traversed" part of line $\overline{st}$. Update face F to be the face which is incident to p and which is intersected by the line segment pt in the region of p and start all over again. There is another method proposed in the same paper [7] (in the original paper the algorithm is called compass routing) and shown to be work for some important classes of networks. Specifically, Kranakis et al. [7] have shown that the compass routing works correctly for Delaunay triangulation. The key idea behind the compass routing algorithm is as follows: If an information packet wants to reach the destination node, t, from the source node, s. Then the algorithm will send the packet to the neighbor u of s such that the slope of the line segment joining u to s is the closest to the slope of the line segment joining s to t. The problem with compass routing is that it may occasionally fall into infinite loops as a result information packet fails to reach the destination. But, this problem never occurs in case of networks whose underlying graph is Delaunay triangulation.

To the best of our knowledge, Bose et al. [8] were the first who presented average-case efficient routing algorithm with guaranteed delivery in ad hoc wireless networks. In fact, these are the first distributed algorithms that do not require duplication of information packets or any kind of duplication of memory at the nodes. Furthermore, the distributed algorithms only require a constant amount of information along with actual message from source node to destination node. Moreover, the paper presented a simulation to measure the quality of the paths described by the algorithms. The authors choose to compare their algorithms with distributed algorithm described by Lin et al. [14]. In my opinion, this choice is quite unusual since the algorithm in [14] fails in the case when the information packet consecutively crosses the same edge of the graph twice. Our work is inspired by and based upon the idea articulated in this paper [8]. That is, the idea of finding a planar graph and then applying routing algorithm on this extracted graph.

To the best of our knowledge, the first worst-case analysis of geometric mobile ad hoc routing was presented by Kuhn et al. [15]. In particular, Kuhn et al. presented first adaptive face routing algorithm with the cost bounded by a function of the optimal route. For example, if a best route has a cost c (say), then the adaptive face routing finds a route and terminates with cost $O(c^2)$ in the worst case. In addition, they also gave a tight lower bound by showing that any geometric routing algorithm has worst-case cost $\Omega(c^2)$. The lower bound analysis presented in the present work is based on the idea presented in Kuhn et al. [15].

Other geometric routing algorithms [8], [16], that guaranteed to find the destination on triangulations or convex subdivisions [17], have been suggested. The algorithm in [18] used the Delaunay triangulations as an underlying graph to find the destination node. Consequently, local approximation of Delaunay Graph has been suggested [19], providing however, no better bound on the performance of routing algorithms in [8], [16]–[18].

The first algorithm that is competitive with respect to the shortest path between the source and the destination was proposed in [15]. This algorithm basically enhances Face Routing by the concept of a bounding ellipse restricting the searchable area. The lower bound example was shown in [15] to be asymptotically optimal.

A more detailed discussion of geometric routing can be found in [20].

III. COMPUTATIONAL MODEL

This paper considers the standard model for ad hoc networks where all nodes have the same transmission range. That is, in this paper we model an ad hoc network as a geometric graph. A graph G is define as a pair (N, E), where a set N representing the nodes of the network and a set $E \subseteq N^2$ denotes the set of edges, connecting the nodes in N. Furthermore, there exists an edge between two nodes u and v if, and only if, the nodes u and v are within mutual transmission range. In the remainder of this paper we will refer to the elements of N alternately as host, nodes, or vertices. Since, we are assuming that all nodes have equal broadcast ranges. This leads to the definition of the unit disk graph (UDG). Let N be a set of points in the plane. Then the UDG is a geometric graph that contains a vertex for each element of N. An edge (u, v) is present in $UDG(S)$ if, and only if, $dist(u, v) \leq 1$ (this can be accomplished by scaling the coordinate system), where $dist(u, v)$ denotes the Euclidean distance between x and y. Now if we combine the above definitions of graph and unit graph; and generalize the statement, we have a graph G where each node has a coordinate in 2-dimensional space. More formally, a 2-dimensional embedding of G is a coordinate function coord: $N \to R^2$ of the nodes. A graph G is a unit disk graph (UGD) if it has an embedding such that the Euclidean distance dist(coord(v), coord(u))$\leq$1 if, and only if, $\{u, v\} \in E$.

IV. PRELIMINARIES AND DEFINITIONS

This paper assumes that the underline graph is geometric graph. A geometric graph $G = (N, E)$ is a graph in the plane so that its vertices are points and its edges are straight-line segments connecting pair of these points. Recall, a geometric graph need not be planar. Note that when it is clear from the context, we refer to the element of N interchangeably as host, nodes, or vertices.

The open neighborhood $\mathfrak{N}(v)$ of the vertex v consists of the set of vertices adjacent to v, that is, $\mathfrak{N}(v) = \{w \in N : vw \in E\}$ and the closed neighborhood of v is $\mathfrak{N}[v] = \mathfrak{N}(v) \in \{v\}$. For a set N, the open neighborhood $\mathfrak{N}(N)$ is defined to be $\cup_{v \in N} \mathfrak{N}(v)$, and the closed neighborhood of N is $\mathfrak{N}[N] = \mathfrak{N}(N) \cup N$.

The key to the Face Routing algorithm [7] is the right hand rule. The right-hand rule [21] states that when arriving at node x from node y, the next edge traversed is the next one sequentially counterclockwise about x from edge (x, y). It is known that the right-hand rule traverses the interior of a closed polygonal region (a face) in clockwise order and traverses an exterior region in counterclockwise edge order.

The Gabriel graph, denoted by $GG(N)$, of set N is defined by the graph in which uv is an edge of $GG(N)$ if, and only if, the circle having $\|uv\|$ as a diameter is an empty circle, that is, if and only if it contains no points of N in its interior. $GG(N)$ can be computed in $\Theta(n \log n)$ time by removing edges $DT(N)$ edges not intersecting their Voronoi edges. Note that $GG(N)$ is a subgraph of $DT(N)$ and contains at least one MST for N. For details reader may consult [22]. Unit Gabriel graph, $GG(V)$, consists of all edges such that $||uv|| \leq 1$ and the circle having diameter uv does not contain any vertex from V and the lengths of edges not more than unity.

The relative neighborhood graph, denoted by $RNG(N)$, is defined as a geometric graph in which $RNG(N)$ has an edge between n_i and n_j if and only if $d(n_i,\ n_j) = min_{k \neq (i,j)}\ max(d(n_i,\ n_k),\ d(n_j,\ n_k))$. $RNG(N)$ can be computed in $\Theta(n \log n)$ time. Furthermore, $RNG(N)$ is a subgraph of $DT(N)$ and contains at least one MST for N.

For a given set N of n points, Delaunay Triangulation, denoted by $DT(N)$ is defined as the unique triangulation in which the circumcircle of each triangle does not contain any other points of N in its interior. Delaunay Triangulations can be computed in $\Theta(n \log n)$ time and contains at least one minimum spanning tree (MST) for a given set N. Note that MST for N can be computed in $\Theta(n)$ time once the Delaunay triangulation of N is given [23].

The straight-line dual of the Voronoi diagram for N is a triangulation of N, known as Delaunay triangulation and denoted by $DT(N)$. Also, defined as the unique triangulation in which the circumcircle of each triangle does not contain any other point in its interior. Delaunay Triangulations can be computed in $\Theta(n \log n)$ time and contains at least one minimum spanning tree (MST) for the given set N. The MST for N can be computed in $\Theta(n)$ time once Delaunay triangulation of N is given [23]. The k-localized Delaunay graph, $LDel^{(k)}(N)$, on the nodes set N has unit Gabriel edges and edges of all k-localized Delaunay triangles. Li et al. [24], [25] defined the k-localized Delaunay triangle as the triangle if the interior of the circumcircle of the triangle does not contain any node of N that is a k-neighbor of any of three nodes of the triangle. Furthermore, all edges of the triangle have no more than a unit length. Li et al. [24], [25] have shown that the graph $LDel^{(1)}(N)$ may contain some intersecting edges but the graph $LDel^{(2)}(N)$ contains no intersecting edges i.e. planar. Hence the following lemma.

Lemma 1: For any $k \geq 2$, The k-localized Delaunary graph, $LDel^{(k)}(N)$, is a planar graph [24], [25] .

For proof the Reader may consult [24], [25].

Since our algorithm is based on the construction of localized Delaunay triangulation presented in [24], [25] and for completeness we outline the method here. After gathering the identity and location information by the nodes, the algorithm finds the Gabriel edges as follows:

each edge uv is a Gabriel edge if both angles opposite to uv are less than $\pi/2$. Each node finds all triangles such that all three edges of the triangles have at most one unit length. If the angle at the node is at least $\pi/3$, node u proposed to form the Delaunay triangulation. If two nodes either accept the proposal or send the proposal, the edge uv is in the Delaunay triangulation. Li et al. [24], [25] have shown that in each triangle one of its interior angle is at least $\pi/3$ and form a 1-localized Delaunay triangle. In addition, they showed that if angle at the node is at least $\pi/3$, then the node broadcast at most 6 proposals to construct Delaunay triangulation. The total cost of this construction is $O(n \log n)$ bits. For details and associated proofs, readers may consult [24], [25].

V. Planarization and Face Traversal

This section discusses the conditions responsible for the failure of planar graph and/or routing. Even though the conditions discussed here is from purely practical perspective, but we believe that these condition are important in the algorithm design, for after all, the eventual goal of any algorithm (protocol) is the implementation in the real life conditions such as presence of obstacles, variable communication ranges, environmental condition etc. Like all geometric/face routing algorithms [8], [26], [27], [29], our geometric algorithm depends upon two factors namely, construction of planar subgraph and face traversal mechanism. Planar subgraph defines the underlying wireless connectivity graph while the face traversal defines a forwarding mechanism for information packet from source to destination i.e. routing.

The paper considered the unit-disk graph model because an underlying assumption of the planar construction algorithm and face traversal algorithm is that this computational model defines the connectivity among nodes in ad hoc wireless network. That is, any violation of the unit-disk graph assumption results in failure either in the construction of planar subgraph or in the face traversal mechanism. As a result, geometric routing algorithm fails to deliver packet to its destination. We define the “failure” in context of geometric routing as the algorithm’s lack of ability to find a path for at least one pair of nodes between source and destination. Note that there are some studies [28] in which unit-graph assumption is relaxed. The technique presented in those works is scalable only when certain restrictions were applied to the radius of communication nodes. Moreover, the technique deals with virtual links.

In general, our geometric routing algorithm relies on three distinct components. Firstly, our algorithm uses greedy forwarding paradigm. That is, a node under consideration sends a packet to its neighbor node that is geometrically closest to the destination node. Secondly to traverse the face, our algorithm uses right-hand rule and face changes. Note that the important aspect of the right-hand rule is that it must supplement with a rule that clearly defines when to change face. Our algorithm follows the suggestion of Bose et al. [8]. Thirdly, the construction of planar subgraph. The primary reason of this construction is that geometric routing provably works correctly on a network graph that has no crossing links, i.e. a planar graph. In addition, planar construction of subgraph eliminates crossing links to generate the structure that is required for the face traversal. In the literature, geometric routing algorithms construct planar subgraph using proximity structures from computational geometry particularly, the Gabriel Graph (GG) [30], the Relative Neighborhood Graph (RNG) [31], or Restricted Delaunay Graph (RDG) [19]. The reason for the use of these graphs is that these proximity structures provably yield a connected, planar graph provided the connectivity among the nodes obeys the unit-disk graph assumption.

Taking account of all three components of geometric algorithms, we could generalize the overall structure of our work as follows.

1) Construct the planar subgraph from the graph represented the given set of nodes.

2) A node sends packet to its geometrically closest neighbor until it reaches to the destination.

3) If there exists no such neighbor i.e., greedy traversal fails.

4) Then invoke the greedy (face) traversal algorithm (right-hand rule is implicit in this step) to recover from failure.

5) If face traversal come across a node closer to the destination.

6) Then recover from failure by falling back to greedy traversal i.e. Goto Step 2.

In the remainder of this section, we discuss some major reasons, which lead to the failure in the construction of planar subgraph. From the practical viewpoint, the unit-disk graph assumption fails two circumstances. Firstly, the node may estimate its position incorrectly. Secondly, the presence of obstacles, which causes the regularity in the communication ranges. These two practical circumstances cause the best geometric/phase routing algorithms [7], [8], [26], [27], [29] in severe degrading in the performances, even responsible for failure in some instances. The deviations from ideal behavior can cause three kinds of failures in the construction of planar subgraph.

One of the most important consequences of the node’s incorrect estimation of its position is that it can leave unidirectional links in the derived subgraph [29], which is usually Gabriel graph or relative neighborhood graph. Particularly, At least one of the node x (say) does not know the existence of at least one node y (say) in the communication graph and node x keeps link with some other node that should not be there in the resultant subgraph. On the other hand, other node removes the same link. It is easy to see that incorrect estimate of node’s position leads to these unidirectional links. Furthermore, these unidirectional links can produce an infinite loop during face traversal. Using the same line of arguments, we can show that the algorithm may leave the cross-links in the resultant subgraph. References [29], [32] discuss incorrectly removal of links and leaving the cross-links in the subgraph in their simulation studies.

The presence of obstacles can also cause a link to be incorrectly removed from the resultant subgraph, which clearly results in a disconnected subgraph. As we have mentioned above, a disconnected resultant subgraph causes a face traversal failure. By failure in the traversal we mean that greedy traversal cannot reach certain nodes, since the resultant subgraph is partitioned. Again, the similar line of arguments can show that crossing links in the resultant subgraph can cause persistent routing failures. In this case, the failure is usually due to the infinite loop.

VI. Overall Strategy of the Algorithm

For the ease of the presentation, the overall strategy of the algorithm can be divided in two distinct phases. The phase I extracts the connected and planar graph from the given graph while phase II does the actual routing on the graph produced by phase I.

In the phase I, we proposed the distributed algorithm, which is based on the [24], on unit disk model. The basic idea of the algorithm is as follows. Each node in the given graph, G, broadcast its identity and position (coordinates) and; gather identities and positions of their neighbor nodes. Using this information, each node u computes the local Delaunay triangulation, LDG, such that edges of the triangles are not larger than one unit. This part of the algorithm is based on the distributed algorithm proposed in [24]. Now each node sends the message to its neighboring node to remove the edges which are not Gabriel edges. When node u receives a message REMOVE(edge(u, v)), it accepts if there is no point (some node) lies in the disk of diameter $\|uv\|$ otherwise, rejects it by sending the message REJECT(edge(u, v)). For example, if node u sees that there is a point (some node) in the intersection of its closed neighborhood, $\mathfrak{N}(u) \cup \{u\}$, and the disk of diameter $\|uv\|, disk(u, v)$, then node u sends the message to node v (along with other neighbors) to remove the edge (u, v), REMOVE(edge(u, v)). Similarly, node v sends the message to remove edge (v, u) to node u (along with other neighbors), if it sees that there exists some node in $(\mathfrak{N}(v) \cup \{v\}) \cap disk(v, u)$. If u and v both sent and received the message REMOVE, then the edge (u, v) will be removed. In other words, if node u has sent the remove message to node v and also received the remove message from node v, then the edge (u, v) is removed from the local Delaunay graph, LDG. Since DT is planar and GG is connected, therefore, the graph produced by the algorithm is planar and connected.

In the phase II, we will run the routing algorithm based on the Face algorithm [7] (see Section II for detailed description) on the graph computed by the algorithm in the phase I.

VII. Planar Graph Construction

This section presents the fully distributed algorithm to extract the connected, planar graph. This algorithm also appeared in [5] and the informal version appeared in Reference [6]. The idea of the construction of the planar subgraph is based upon the [24]. Li et al. [24] proposed a localized algorithm that constructs a sequence graphs, called localized Delaunay $LDel^{(k)}(N)$. Our proposed algorithm starts with Li et al. localized algorithm and systematically removed the edges such that the resultant subgraph is connected. We formalize the idea presented in section VI in the distributed algorithm as follows.

Algorithm: Connected Graph

1. Each node u broadcast its identity and listens to the messages form other nodes.
2. Each node u computes the DT with its neighbors (with $\mathfrak{N}[u] = \mathfrak{N}(u) \cup \{u\}$)
3. For each edge uv, such that disk with diameter $\|uv\|$ contains no other points of N. Let $\triangle uvw$ and $\triangle uvz$ be two triangles with common edge uv.
4. For each edge $u \in \mathfrak{N}(u) \cup \{u\}$
   4.1. Find all triangles $\in Del(\mathfrak{N}(u) \cup \{u\}$ such that $\|uv\|$, $\|vw\|$, $\|wv\| \leq 1$.
   4.2. IF disk$(u, v) \cap ((\mathfrak{N}(u) \cap u) \setminus \{u, v\}) \neq \{\}$, THEN
      4.2.1. Node u broadcast a message PROPOSAL(edge(u, v)) to remove edge(u, v) from Delaunay triangle $\triangle uvw$.
      4.2.2. Node u listens to the message from other nodes.
5. When node v receives a message PROPOSAL(edge(u, v))
   5.1. Node v checks the existence of some node in its closed neighborhood i.e.,
      5.1.1. IF disk$(u, v) \cap ((\mathfrak{N}(v) \cup v) \setminus u, v) \neq \{\}$ THEN
      5.1.2. Node u broadcast the message ACCEPT(edge(u, v)) //u accepts the proposal to removing edge(u, v).
      ELSE
      5.1.3. Node u broadcast the message REJECT(edge(u, v)) //u rejects the proposal to removing edge(u, v).
6. IF edge$(u, v) \in Del(\mathfrak{N}_1(u) \cup \{u\})$ AND [u and v sent REJECTS(edge(u, v))] THEN
   6.1 Node u deletes the edges (u, v) from its set of incident edges.

The basic technique here is to find the intersection of UDG and local Delaunay triangulation (see lines 4.2 and 5.1.1). This technique was proposed in [8] in which Bose et al. used Gabriel graph with unit graph and showed that minimum spanning tree is a subgraph of unit graph. Since, $MST(N)$ is a subset of $DT(N)$, we can apply the same technique [8] to show that the resultant subgraph is connected if the intersection of unit disk graph and the graph produced by the above algorithm is connected. Hence the following lemma.

Lemma 2: If the unit disk graph of the given set, $UDG(N)$ is connected, then $LDel(N) \cap UDG(N)$ is connected.

Proof:

Let's take the negation of the lemma and suppose it to be true. That is, there exists an edge $(u, v) \in MST(N)$ such that $\|uv\| > 1$. Removing the edge(u, v) form $MST(N)$ will splits the graph into two connected components C_1 and C_2 such that $u \in C_1$ and $v \in C_2$. By definition, $UDG(N)$ is connected, it means that there must exists some edge (w, x) with $\| wx \| \leq 1$ such that $w \in C_1(N)$ and $x \in C_2(N)$. If the edge (u, v) is replaced by the edge (u, v), then the resultant graph on N would be connected with weight less than $MST(N)$, a contradiction. Thus, the graph produced by the algorithm is connected. ■

The following lemma, which is based on [24], establishes the relation between the subgraph produced by the algorithm and the Gabriel graph i.e. the edges computed by the algorithm is in fact Gabriel edges.

Lemma 3: If the edge (u', v') computed by the algorithm intersects a local Delaunay triangle (u, v, w), then either u' or y' is inside the disk (u, v, w).

Proof: Let triangle (u, v, w) be a triangle with circumcenter c, then by definition, at least one of the u, v, or w must be on the different side of the line $u'v'$ with center (say u). If both u' an v' are outside of circum circle defined by (u, v, w), then $\angle v'uu' > \pi/2$. Therefore, u is inside the disk (u, v), which is contradicts that $u'y'$ is a Gabriel edge. ■

Now we show that the graph produced by the algorithm is planar.

Lemma 4: The Algorithm Connected Graph produced the planar graph G.

Proof: Assume that there exists an edge (x, y) such that (x, y) intersects triangle (u, v, w). Now, edge xy is either a Gabriel edge or an edge of a Delaunay triangle. If xy is a Gabriel edge, then either x or y is inside the disk (u, v, w). On the other hand, if xy is an edge of a Delaunay triangle, then either x or y is inside the disk (u, v, w). In any event, there exists one endpoint (either x or y) with is outside of disk(u, v, w). This means that x (say) is inside the closed neighborhood, $\mathfrak{N}(u) \cup \{u\}$, and y (say) outside the closed neighborhood, $\mathfrak{N}(u) \cup \{u\}$. This implies that there is a path from side the closed neighborhood, $N(u) \cup \{u\}$, to the outside of the neighborhood (since algorithm produces connected graph), a contradiction. Thus, there is no edge in the resultant graph that intersects triangle (u, v, w). Since we have no intersecting edges in the resultant graph, the graph is planar by definition. This completes the proof. ■

Although, the above a lemma showed independently that the resultant subgraph produced by the algorithm is planar, we can deduced the planarity of the resultant subgraph from Lemma 2 as follows. Since UDG is planar and the local Delaunay triangulation is planar, therefore, the intersection of UDG and Delaunay triangulation must be planar.

The subsequent line of arguments is based on the Li et al. [24], for brief introduction see Section IV. Since a graph produced by the algorithm is planar by Lemma 4, and a proposal made only if $\angle wuv \geq \pi/3$, node u broadcasts at most 6 proposals. And each proposal is replied by at most two nodes. Therefore, the total communication cost is $O(d \log d)$ bits, where d is the degree of a node.

VIII. Geometric Routing

In this section we describe an algorithm for routing in planar graph obtained in the previous section. The informal versions of this algorithm appeared in [5], [6]. The algorithm is based on [7] and the basic idea is as follows. Let f be the face of G with a starting point p on its boundary that intersects line segment (p, t), where t is the destination. Using right-hand rule, traverse the face f in the counterclockwise direction. If the edge (u, v) of the face f intersects with (p, t) at p' and $dist(p', t) < dist(p, t)$ then this intersection p' becomes the new starting point p. In the similar fashion traverse faces until p becomes a destination t. We used the same idea but instead interchanging the points, the traversal moves to the adjacent face crossed by line segment (p, t) as suggested in [8]. Note that even though we have not mentioned it explicitly in the algorithm, the traversal is accomplished by right-hand rule. See Section V for the relation between traversal and right-hand rule .

Algorithm Greedy Traversal

```
p = s
for all edge (u, v) ∈ f do
    if (u, v) intersects (p, t) at a point p' then
        if face f contains p and p = t then
            we are done
        else
            move to the adjacent face
```

For the sake of the argument (adversary argument), we say that the above algorithm might drop the packet if a point, p', for a face change is not found after the complete face traversal. We claim that since underlying graph is planar, this situation will never occur provided the destination is connected. Reader may refer to the Section V for details. In particular, since underlying subgraph is planar by Lemma 4 and the destination is connected (implied by Lemma 2), the face-change rule must always find a face closer to destination than the current face along line segment joining source-destination. See brief introduction of face routing [7] in the Section II.

The subsequent two lemmas establish the correctness by showing the forward progress of the packet and that the algorithm will terminate after executing a finite number of steps.

Lemma 5: Algorithm Greedy-Traversal reaches destination t in a finite number of steps.

Proof: Since the underlying graph is connected and planar by Lemma 4 and since the distance to the destination is decreasing during each pass of the algorithm, therefore, algorithm Greedy-Traversal reaches destination t in a finite number of steps. ■

Following lemma gives the bound on the number of steps.

Lemma 6: Algorithm Greedy-Traversal reaches the destination t after at most $4|E|$ steps, where $|E|$ is the number of edges in G.

The key to the proof of Lemma 6 is that the algorithm must go across the entire face f to determine the point of intersection and then must return to the intersection point. For proof, reader may consult [7]. However, to emphasis the importance of Lemma 6 in the context of the geometric algorithms, we like to add interesting point regarding the bound appeared in the Lemma 6. That is, Bose et al. [8] suggested the way to reduced the bound $4|E|$. They proposed that the bound $4|E|$ of the face routing [7] can be reduced to $3|E|$ by forcing the algorithm to follow the shortest path when returning to intersection points. Although we believe that this "cunning" observation of Bose et al. [8] would increase the performance of any face routing algorithm [7], [8], [14], [20] in practice especially, in the case of present work, but we are not sure what consequence this might have on the algorithmic complexity. Particularly, how this "shortest path" calculation influence the overall complexity of the algorithm. We hope to extend the present work by including this fine point of Bose et al. [8].

IX. Lower Bound Analysis

This section presents a lower bound for the routing algorithm running on the connected planar graph constructed in Section VII. We formalized the reasoning presented in [15] to establish the lower bound of the algorithm.

Lemma 7: The algorithm presented in section has the cost $\Omega(n^2)$ where n is the number of nodes in G.

Proof: Suppose G be the Euclidean graph G and $2k$, where k is positive integer, nodes are evenly distributed on a circle. Since we are working on UDG, it is reasonable to suppose (without loss of generality) that the Euclidean distance between neighboring nodes is exactly 1. There are routes (paths) from the nodes on the boundary of circle towards the node at the center of the circle such that the distance between adjacent nodes is exactly 1. Now, let s be the particular but arbitrarily chosen node on the circle boundary such that there exists a route (path) from node s to the node t at the center. The algorithm finds the route (path) from the source node s to the destination node to at the center. As shown in [15], an optimal route between source and the destination follows the shortest path on the circle until it hits the node s and then directly follows $s's$ path to t. The cost of such link is not larger than the number of nodes in a circle of given radius i.e. $c \leq k+r+1$. In other words, the cost of the optimal path from source s at the circle boundary to destination t at the center is $O(n)$. In this scenario, the goal of the routing algorithm is find the correct path that leads to the destination node t. Since according to our assumption, there is no routing information is stored at the node, the only way to find such path is search exhaustively for the required path. Therefore, the algorithm has to explore n^2 nodes in the worst case. Because the underlying graph is unit disk graph with edge length exactly 1 (according to our assumption) leads to the cost $\Omega(n^2)$. This completes the proof. ■

We have shown that in the worst the case our algorithm is quadratic in the cost of an optimal path in the given graph G.

X. Conclusion and Outlook

We have presented a technique to extract the connected, planar subgraph for geometric routing algorithms. We considered the idealized unit disk graph model in which nodes are assumed to be connected if, and only if, nodes are within their transmission range. The main contribution of this paper is a fully distributed algorithm to extract the connected, planar graph for routing in the wireless networks. The communication cost of the proposed algorithm is $O(d \ log \ d)$ bits, where d is the degree of a node. We have also presented the geometric routing algorithm that is based upon the famous Face Routing algorithm. The algorithm is fully distributed and nodes know only the position of other nodes and can communicate with neighboring nodes in their transmission range. We have shown that in the worst case our algorithm is quadratic in the cost of an optimal path in the given graph.

Mobility is one of the most important features of an ad hoc wireless networking, but we have limited the scope of the present work by assuming that the routing was taking place much faster than nodes movement. The inclusion of node(s) movement is the next logical topic for the future research. In addition, we hope to extend our work beyond the flat topology. A promising direction is to work in 3D-volumes rather than 2D-faces.

References

[1] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, 15(6):30-39, November 2001.

[2] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad hoc networks," in *Proceedings of 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, pp. 102-111, Tampa, Florida, USA, August 2001.

[3] S. Giordano and I. Stojmenovic, "Position based routing algorithms for ad hoc networks: a taxonomy," in *Ad Hoc Wireless Networking*, X. Cheng, X. Huang, and D-Z. Du, Ed. Boston:Kluwer Academic Publishers, pp. 103-136, 2004.

[4] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," *Mobile Networks and Applications*, 1(2):89-104, 1996.

[5] R. B. Muhammad, "A geo-routing algorithm in planar graph for ad hoc wireless networks," in *Proceedings of the 21st IEEE International Conference on Advanced Information Networking and Applications Workshops (AINA Workshops 2007)*, pp. 685-689, Niagara Falls, Canada, May 2007.
[6] R. B. Muhammad, "A distributed geometric routing algorithm for ad hoc wireless networks," in *IEEE Proceedings of the 4th International Conference on Information Technology: New Generations (ITNG 2007)*, pp. 961-963, Las Vegas, USA, April 2007.
[7] E. Kranakis, H. Sing, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of 11th Canadian Conference on Computational Geometry (CCCG 1999)*, pp. 52-54, Vancouver, Canada, August 1999.
[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'99)*, pp. 48-55, Seattle, USA, August 1999.
[9] H. Takagi and L. Kleinrook, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, 32(3):246-257, 1984.
[10] T. C. Hu, and V. O. K. Li, "Transmission range control in multihop packet radio networks," in *IEEE Transactions on Communications*, 34(1):38-44, 1986.
[11] G. Finn, *Routing and addressing problems in large metropolitan-scale internetworks*, Technical Report, ISS/RR-87-180, USC/ISI, Department of Computer Science, University of Southern California, USA, March 1987.
[12] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multiple-hop ad hoc networks", in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 1999)*, pp. 243-254, Seattle, USA, August 1999.
[13] T. Hou and V. Li. "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, 34(1):38-44, 1986.
[14] X. Lin and I. Stojmenovic. *Geographic distance routing in ad hoc wireless networks*, Technical Report, TR-98-10, School of Information Technology and Engineering, University of Ottawa, Canada, December 1998.
[15] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically Optimal geometric mobile ad-hoc routing," in *Proceedings of 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, (DialM 2002)*, pp. 24-33, Atlanta, USA, September 2002.
[16] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," in *Proceedings of IEEE 21st International Conference on Distributed Computing Systems Workshop (ICDCSW 2001)*, pp. 461-466, Phoenix, USA, April 2001.
[17] P. Bose, P. Morin, A. Brodnik, S. Carlsson, E. Demaine, R. Flesischer, J. Munro, and A. Lopez-Ortiz, "Online routing in convex subdivisions" in *Proceedings of the 11th International Symposium on Algorithms and Computation (ISAAC 2000)*, vol. 1969, Springer Lecture Notes in Computer Science, pp. 47-59, Taipei, Taiwan, 2000.
[18] P. Bose and P. Morin, "Online routing in triangulations" in *Proceedings of the 10th International Symposium on Algorithms and Computation (ISAAC 1999)*, vol. 1741, Springer Lecture Notes in Computer Science, pp. 113-122, Chennai, India, 1999.
[19] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proceedings of 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, pp. 45-55, Long Beach, USA, 2001.
[20] J. Urrutia, "Routing with guaranteed delivery in geometric and wireless networks," in *Handbook of Wireless Networks and Mobile Computing*, I. Stojmenovic, Ed. New York:Wiley, pp. 393-406, 2002.
[21] J. A. Bondy and U. S.R. Murty, *Graph Theory with Applications*, New York:American Elsevier, 1976.
[22] B. Boots, A. Okabe, K. Sugihara and S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, New York:Wiley, 2000.
[23] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, New York:Springer Verlag, 1985.
[24] X. Y. Li, G. Calinescu, and P. J. Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, pp. 1268-1277, vol. 3, New York, USA, June 2002.
[25] X. Y. Li, "Applications of Computational Geometry in Wireless Networks," in *Ad Hoc Wireless Networking*, X. Cheng, X. Huang, and D-Z. Du, Ed. Boston:Kluwer Academic Publishers, pp. 197-264. 2004.
[26] B. Karp, and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 243-254, Boston, USA, August 2000.
[27] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad hoc routing: of theory and Practice," in *Proceedings of the 22nd Annual ACM Symposium on Principals of Distributed Computing (PODC 2003)*, pp. 63-72, Boston, USA, July 2003.
[28] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc networks beyond unit disk graphs", in *Proceedings of the 2003 Joint ACM Workshop on Foundations of Mobile Computing (DIALM-POMC'03*, pp. 69-78, San Diego, USA, September 2003.
[29] Y. Kim, R. Govindan, B. Karp, and S. Shenker, "On the pitfalls of geographic face routing," in *Proceedings of the 2005 Joint ACM Workshop on Foundations of Mobile Computing (DIALM-POMC'05)*, pp. 34-43, Cologne, Germany, September 2005.
[30] K. Gabriel, and R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, 18:259-278, 1969.
[31] G. Toussanit, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, 12(4):261-268, 1980.
[32] K. Seada, A. Helmy, and R. Govindan, "Localization errors on geographic face routing in sensor networks," in *Proceedings of the 3rd ACM International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pp. 71-80, Berkeley, USA, April 2004.

Rashid Bin Muhammad was born in Karachi, Pakistan. He is currently a Ph.D. candidate and a teaching fellow of computer science at Kent State University. He received his MS degrees in physics, industrial education, and computer science from University of Karachi (1990), Murray State University (1994), and Kent State University (2003), respectively. He is a member of both ACM and the IEEE Computer Society. His research interests include parallel and distributed computing, computational geometry, graph theory, and algorithmic issues in wireless networking.