Void Traversal for Efficient Non-Planar Geometric Routing*

Thomas Clouser, Adnan Vora, Timothy Fox and Mikhail Nesterenko†
Computer Science Department
Kent State University
Kent, OH, 44242
{tclouser,avora,tfox,mikhail}@cs.kent.edu

Abstract

Geometric routing provides a scalable and efficient way to route messages in ad hoc networks if extensive routing information is unavailable. Such algorithms require a planar graph to guarantee message delivery. The routing techniques for such guarantee usually center around the traversal of planar faces of the graph. However, in realistic wireless networks existing planarization methods, if at all applicable, tend to require extensive local storage or result in suboptimal route selection. In this paper we study an alternative approach of translating the algorithms themselves to be able to route messages over voids in non-planar graphs. We prove sufficient memory requirements for such translations. We then translate several well-known planar geometric routing algorithms and evaluate their performance in both static and mobile networks.

1 Introduction

Motivation. Geometric routing is a promising approach for message transmission in ad hoc wireless networks. Unlike traditional ad hoc routing, geometric routing algorithms have no control messages and only minimal routing tables. In geometric routing, a node does not have to store any residual information after message transmission. Hence, such routing scales well. Geometric routing can be used either as a primary routing method or as a fallback message delivery strategy if the primary routing infrastructure is unavailable. Geometric routing is particularly appropriate for wireless sensor networks [14]. Networked sensors usually have limited resources for routing information, yet many applications for sensor networks involve large scale ad hoc deployments [2].

Realistic radio networks may lead to frequent changes in topology [11, 35]. Such changes make routing table construction, if feasible at all, very expensive. The stateless properties of geometric routing may be quite

*Some of the results presented in this paper are published in [34].

†This research was supported in part by DARPA contract OSU-RF#F33615-01-C-1901 and by NSF CAREER Award CNS-0347485

appropriate for this setting. Furthermore, the nodes in a wireless network may be not be static. For example, the wireless radios may be mounted on vehicles or carried by people [28]. In this case, geometric routing may be attractive since each individual node holds minimal routing information. Therefore, the routing information does not get obsolete as the nodes move [16].

State of the art. In geometric routing, each node knows it own geographic coordinates as well as the coordinates of its immediate neighbors. These coordinates are either obtained from GPS receivers or a localization algorithm [5, 13]. The message sender knows the coordinates of the destination. This information can be obtained from a location service [1, 26, 30]. Landmark positions instead of precise node locations may also be used for geometric routing [29].

The simplest version of a geometric routing algorithm is *greedy* routing [8]. In such routing, the source, as well as each intermediate node, examines its neighboring nodes and forwards the message to the one that is the closest to the destination. Unfortunately, the greedy routing algorithm fails if one of the intermediate nodes is *local minimum*: all its neighbors are further away from the destination than the node itself. In *compass* routing [20], the next hop is selected such that the angle between the direction to the next hop node and to the destination is minimal. However, even on planar graphs compass routing may fail due to livelock [20].

Face routing complements greedy routing by guaranteeing message delivery in case a local minimum is encountered. It requires that the communication graph is planar. A source-destination line intersects a finite number of faces in this graph. A message reaches the destination by sequentially traversing these faces. One such routing scheme that is frequently used is Compass Routing II [20], also known as *FACE*. *GFG/GPSR* [4, 16] combines greedy and face routing. The algorithm uses greedy routing unless it encounters local minimum. The algorithm then uses face routing to get closer to the destination and switches to greedy routing again. *GFG* assumes that the original communication graph is a *unit-disk graph*: two nodes share an edge if and only if they are within a certain unit-distance from each other. For this graph, the authors construct its Gabriel [10] or relative neighborhood [33] subgraphs. These subgraphs are planar, they preserve the connectivity of the original graph and they can be constructed locally by each node. Datta et al [7] propose to use the edges that do not belong to the planar subgraph if it shortens the route. Recently, more efficient approaches of face traversal on the basis of α-shapes have been proposed [27, 31].

Kuhn et al [21, 22, 24] present a face routing algorithm *OAFR* that, in combination with greedy routing, produces the worst case asymptotically optimal geometric routing algorithm *GOAFR+*. They compare the performance of multiple geometric routing algorithms and demonstrate that in the average case *GOAFR+* also has the best performance. Kim et al [18] discuss challenges of geometric routing. Frey and Stojmenovic [9] address some of these challenges and discuss different approaches to geometric routing. Clouser et al [6] describe an algorithm that achieves fast message delivery in geometric routing through concurrent face traversal. Stojmenovic [12] provides a comprehensive taxonomy of geometric routing algorithms.

Kuhn et al [21, 24] develop an extensive evaluation model for geometric routing algorithms. They compare the performance of the algorithms with respect to the graph density. Their findings indicate that for planar graphs, the *critical density* range of the graph is between 3 and 7 nodes per unit-disk. Outside this range, the geometric routing algorithm optimization is immaterial: below 3 nodes per unit-disk the graphs are too sparse to have a large number of alternative routes, while above 7 the graphs seldom have local minima and the greedy routing almost always succeeds. However, within this range, the paths selected by the geometric routing algorithm may substantially differ from optimal and this is where performance optimization has the greatest impact.

Non-planar routing. One of the major shortcomings of traditional geometric routing is the need to planarize the communication graph to be able to recover from local minima. This results in eliminating the intersecting edges. Yet, since the eliminated edges are not used for routing decisions, the resultant routes may be suboptimal. Furthermore, such planarization can be cone efficiently only for unit-disk graphs: two nodes are connected if and only if they are within a certain unit distance. To eliminate extraneous edges in such graphs, it is sufficient for each node to independently examine the location of its neighbors. However, the connectivity of practical radio networks hardly allows this kind of regularity [11, 35]. Barrière et al [3] and Kuhn et al [23] consider a more realistic *quasi-unit-disk* graph model. In this graph, for each node, outside the disk of definitive connectivity, there is a band where the connectivity is probabilistic.

Kuhn et al [23] reconstruct a unit-disk graph out of a quasi-unit-disk graph. For each edge that should be in a unit-disk graph but is missing from the quasi-unit-disk graph, they add a virtual link that is simulated by a path between the adjacent nodes. This reconstructed graph can then be planarized and used for face routing. However, the start up costs of construction and planarization may be significant. Another planarization approach was suggested by Kim et al [17]. They propose a Cross-Link Detection Protocol (CLDP). The idea is to probe the neighboring nodes and eliminate crossing edges. This approach is coupled with creating Gabriel subgraph to make the communication graph sparser and more efficient for routing. To ensure that the graph is not disconnected, each node may have to be probed multiple times. This protocol is shown [25] to have poor message cost. To overcome this, Kim et al [19] introduce a version of their protocol that only removes cross-links that induce a loop. That is, it is a reactive protocol that is run upon the introduction of the cross link upon the impacted area. However, this introduces routing infrastructure and the complexity of the protocol may still be prohibitively high for some applications. Leong et al [25] propose to divide the network into multiple convex hulls, maintain a hull tree inside each hull and route the message over such trees. However, depending on the communication graph, the discovery and maintenance of these trees requires some nodes to communicate to geometrically distant nodes. That is, this approach may result routes that are far from optimal.

In general, the direct application of classic planarization techniques may result in significant startup costs as the planar subgraph has to be computed without regard to the particular route to be used.

Our contribution. In this paper study an alternative method of navigating a non-planar graph. The idea is to identify and traverse edge segments that border voids of a non-planar graph as if they were faces of a planar graph: the edge intersections of the non-planar graph are treated as virtual nodes while the routing decisions for these virtual nodes are done at real nodes. This way, any planar geometric routing algorithm can be run on a non-planar graph. Kuhn et al mention this idea in their paper [23].

There may be multiple ways to implement this idea and translate a planar routing algorithm to a non-planar. For example, a message may be sent around the area to gather the necessary routing information. In this paper we focus on *direct translation*: each virtual node has a single designated real node that makes the routing decision for the virtual node and forwards the message to the next designated real node. To accommodate this translation, real nodes need sufficient information about their graph neighborhoods. This information is stored in a neighborhood relation.

In this paper we prove that *two-hop completeness* is sufficient condition for the neighborhood relation to direct translate an arbitrary planar geometric routing algorithm. We prove that more limited *one-hop completeness* is sufficient to direct translate *FACE*, *GFG*, *OAFR* and *GOAFR+*.

We then evaluate the performance of the translated algorithms. For that, we recreate the performance evaluation model of Kuhn et al [21, 24] and modify it for non-planar graphs. Specifically, we use a quasi-unit disk rather than simple unit-disk radio communication model. Using our technique, we port the two major planar face traversal algorithms: *FACE* and *OAFR* to operate on non-planar graphs. This results in two new non-planar face routing algorithms: *VOID* and *VOAFR* respectively. Adding greedy routing, we create complete non-planar geometric routing algorithms: *GVG* and *VGOAFR+*. We evaluate the performance of these algorithms in “raw” quasi-unit-disk graphs and in Gabriel subgraphs of these graphs. We then consider the case of mobile networks. We implement random waypoint mobility model [15] and evaluate the performance of *VOID* and *GVG* while varying node speed.

Paper organization. This paper is organized as follows. We describe our notation in Section 2. We introduce one- and two-hop closure of neighborhood relations and prove that these relations are sufficient for algorithm translation as well as give an example of such translation in Section 3. In Section 4, we present the performance evaluation of the non-planar geometric routing algorithms. We conclude the paper in Section 5.

2 Preliminaries

Graphs, faces, voids. We model a wireless network as a geometric graph. A geometric graph $G = (V, E)$ is a set of *nodes* (*vertices*) V on a Euclidean plane connected by *edges* E. The number of nodes is $n = |V|$. Denote $(u, v) \in E$ an edge between nodes u and v. If there is an edge (u, v) then nodes u and v are *adjacent*.

Denote $path(u, v)$ a path from u to v in G. We only consider connected graphs, meaning that for any pair of nodes u and v, the graph G contains $path(u, v)$. Graph G is *planar* if its edges intersect only at nodes. *Void* is a region on a plane such that any two points of this region can be connected by a curve that does not intersect any of the edges of the graph. A void in a planar graph is *face*. A boundary of a void contains the segments of the edges bordering the void. In every finite graph there is one unbounded *external* void. The border of a face forms a simple cycle.

Each edge of a planar graph borders at most two faces. An edge of a non-planar graph may contain the segments of the borders of arbitrarily many faces.

Planar geometric routing. *Source* s and *destination* d are an arbitrary pair of nodes in a graph. The source has a message to transmit to the destination. The source knows the coordinates of the destination.

The payload of the message is immaterial. A routing algorithm specifies a procedure for intermediate node selection. A *geometric routing algorithm with guaranteed delivery* ensures that the message eventually reaches the destination with the following constraints: each node u receiving the message, selects the next node only on the basis of the information $N(u)$ about the graph that is stored at the node and the contents of the received message; the message size is independent of the network size; once the message is forwarded, the node does not store any information about it; at any given time there can only be a single packet for each message en route between the source and destination.

A geometric routing algorithm usually consists of two modes: (i) greedy routing, where each node forwards the message to its neighbor that is closest to the destination; (ii) local minimum recovery mode, applied when a node discovers that it is closer to the destination than all its neighbors. Geometric routing algorithms tend to differ by their local minimum recovery modes.

One of the most widely known algorithms is called *GFG/GPSR* [4, 16]. In the sequel we refer to this algorithm as just *GFG*. For a detailed algorithm description refer to Datta et al [7]. Refer to Figure 1 for illustration of the operation of the algorithm. The local minimum recovery mode of *GFG* is a face routing algorithm we call *FACE*. The idea is for the message to sequentially traverse the planar faces of G that intersect the *sd*-line.

The face traversal follows either *left-* or *right-hand rule*. In the left-hand rule, when the message arrives at some node v from node u, v searches for the adjacent node w such that (v, w) is the first after (u, v) counter-clockwise. That is, for any edge (v, x), the angle $\angle uvx$ is no larger than $\angle uvw$. Node v forwards the message to w. An internal face is thus traversed clockwise, the external one — counter-clockwise. In right-hand-rule, the neighbor selection proceeds in the opposite direction. *FACE* may use left- as well as right-hand rule face traversal direction. The initial face traversal direction is determined by the source. The source selects the adjacent node whose incident edge forms the smallest angle with the *sd*-line and forwards the message there. As the message traverses the face, it may encounter the *sd*-line. At this point, the message changes faces. The node that effects the face change is the first node past the *sd*-line in the old face. After the face change, the traversal direction changes. Thus, a message in *FACE* traverses faces that intersect the *sd*-line in the order of the intersection. Since there is a finite number of such faces, *FACE* guarantees eventual delivery of the message to the destination node.

In Figure 1, the message starts from node s, traverses face V_1 clockwise and visits nodes b, c, h, e and m.

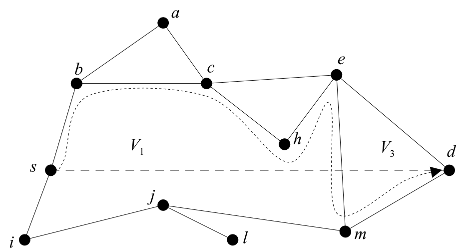

Figure 1: *FACE* example routing. The dotted line is the route taken by the message

At node m, the message changes faces and switches its traversal direction to counter-clockwise. In the next hop the message reaches the destination node d.

To accelerate message delivery, *GFG* starts in a greedy mode and switches to *FACE* only when it encounters a local minimum. To ensure lack of livelocks and guarantee message delivery, *GFG* proceeds in the *FACE* mode until the message is closer to the destination than this local minimum. Then, the message switches back to the greedy routing mode.

Face routing algorithms such as *FACE* may potentially select a rather circuitous route even though a shorter one is available. Suppose a message may reach the *sd*-line significantly faster in one traversal direction than in the other. This is often the case if the *sd*-line intersects the external face. Then, an inopportune selection of the traversal direction results in a suboptimal route. A face routing algorithm *OAFR* [24] mitigates the long route selection by defining an ellipse around the source-destination pair that the message should not cross. If the message traverses a face and reaches the boundary of the ellipse, the message changes the traversal direction. Algorithm *GOAFR+* [21] combines *OAFR* with greedy routing. The resultant algorithm is the most efficient known geometric routing algorithm to date. It outperforms *GFG* on average and its worst-case performance is asymptotically optimal.

Non-planar concepts. A graph is *quasi-unit-disk* with probability p and inner radius r if every two nodes u and v are (i) definitely adjacent if $|u, v| \leq r$; (ii) adjacent with probability p if $r < |u, v| \leq 1$; (iii) definitely not adjacent if $|u, v| > 1$.

A node w is a *mutual witness* [32] for a pair of nodes u and v if the circle whose diameter is (u, v) contains w and w is adjacent to both u and v. A subgraph is *Gabriel* if any two nodes u and v are adjacent only if they do not have a witness. A Gabriel subgraph of any connected graph is connected. The attractive feature of a Gabriel subgraph is that it can be efficiently computed locally by each node just by examining its adjacent nodes. A Gabriel subgraph of a unit-disk graph is planar. A Gabriel subgraph of a quasi-unit-disk graph may not necessarily be planar [3]. However, it can still be effectively computed and it preserves the connectivity of the original graph. Since Gabriel subgraph has fewer edges than the original "raw" graph, the storage and processing demands at each node for routing on such subgraph may be lower.

A segment that borders a void has two incident virtual nodes. A segment belongs to a edge that has two incident real nodes. Real and virtual nodes may coinside. In case the void is traversed, a *head virtual node* is the first incident node of the segment encountered during the traversal. Correspondingly, a *tail virtual node* is the second virtual node. A *head real node* is the node incident to the edge containing the segment such that head real and virtual nodes are on the same side of the traversed segment. *Tail real node* is similarly defined. See Figure 3 for an illustration of these concepts. In this figure, virtual nodes b_2 and b_3 incident to the segment that borders the traversed void while b_1 and b_4 are real nodes incident to the edge that contains this segment. Due to the traversal direction, b_1 and b_2 are respectively real and virtual head nodes while b_4 and b_3 are real and virtual tail nodes.

Note that the tail virtual node of a segment is the head virtual node of the next segment. Also note that a segment borders two voids. Depending on the traversal direction of these two voids, the virtual head nodes either coinside or are on the opposite ends of the segment. The same applies to virtual tail nodes, as well as real head and tail nodes.

Neighborhood $N(v)$ of a node v is the subset of G that includes v. We assume that $N(v)$ is stored at v to let v make routing decisions. In other words, $N(v)$ is the routing table of v. The neighborhood of v may be as small as the set of nodes adjacent to v or as large as the entire graph G. A *neighborhood relation* $\mathcal{N}(G)$ over graph G is an assignment a neighborhood $N(v)$ to each node v of G.

3 Constructing Non-Planar Algorithms

Intersection closure. To enable non-planar geometric routing, neighborhood relations need to have specialized properties.

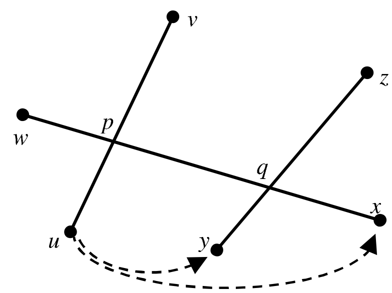

Figure 2: If the neighborhood relation is two-hop closed, the neighborhood $N(u)$ of each node u should contain edges incident to u such as (u,v); edges that intersect (u,v) such as (w,x); edges that intersect (w,x) including (y,z) as well as paths to edges (w,x) and (y,z).

Definition 1 *A neighborhood relation $\mathcal{N}(G)$ over a geometric graph G is* incident edge intersection one-hop closed *(or just* intersection one-hop closed*), if edge (u,v) intersects (w,x) then (u,v), (w,x) and path(u,w) belong to $N(u)$.*

Definition 2 *A neighborhood relation $\mathcal{N}(G)$ over a geometric graph G is* incident edge intersection two-hop closed *(or just* intersection two-hop closed*), if (u,v) intersects (w,x) and (y,z) intersects (w,x), then (u,v), (w,x), (y,z), path(u,w), path(u,y) belong to $N(u)$.*

See Figure 2 for the illustration of Definition 2.

When it is clear from the context we say that the relation is *intersection closed* to mean either or both of the closures of Definitions 1 and 2.

The attractive feature of intersection one-hop closed neighborhood relation is that for every node u and for each incident edge (u,v), the information about every intersecting edge (w,x) , as well as how to reach the nodes incident to this edge is contained in the neighborhood $N(u)$. In case of two-hop closed relation, the neighborhood $N(u)$ contains the information as to how to reach the nodes incident to the edges that intersect the edges incident to u.

The two-hop intersection closed neighborhood relation contains more routing information than one-hop relation. However, the one-hop relation may use less memory resources. If some relation $\mathcal{N}(G)$ is two-hop intersection closed, it is also one-hop closed.

If graph G is planar, its edges intersect only at nodes. Hence, if for each node v, $N(v)$ contains all edges incident to v, then $\mathcal{N}(G)$ is intersection closed according to both definitions. This is not generally the case for non-planar graphs. However, for every connected graph G, there exists an intersection closed relation. Indeed, the neighborhood relation is trivially intersection closed if for each v, $N(v) = G$. That is, each node stores the entire graph G.

A generic intersection closed relation may not be suitable for geometric routing. For example, the size of the path between two nodes incident to intersecting edges may potentially be proportional to the size of the graph. To route a message between these two nodes, the message may have to carry such path. Therefore, the message size may not be constant which violates geometric routing conditions. This motivates the following definitions.

Definition 3 *A neighborhood relation $\mathcal{N}(G)$ is* constant *intersection closed if $\mathcal{N}(G)$ is intersection closed and there is a constant c such that for any graph G, for every node u and every node $v \in N(u)$, there is a $path(v, u) \in N(v)$ whose length is no greater than c.*

Definition 4 *A neighborhood relation $\mathcal{N}(G)$ is* greedy-navigable *intersection closed if $\mathcal{N}(G)$ is intersection closed and for every node u and every node $v \in N(u)$, the greedy routing path $path(v, u)$ belongs to $N(v)$.*

If the neighborhood relation is constant then any node in $N(u)$ can be reached from u in a fixed number of hops. If the neighborhood relation is greedily-navigable, then a message being sent from node u to any node $u \in N(v)$, does not have to carry the entire route. Instead, the message can arrive to u using greedy routing. Hence the following proposition.

Proposition 1 *If an intersection closed neighborhood relation $\mathcal{N}(G)$ is constant or greedy navigable, then there is a constant c such that for any graph G, every node u, and every node $v \in N(u)$, a message can be routed from u to v with its size never exceeding c.*

A constant neighborhood relation may not be navigable and vice versa. In general, it may not be possible to construct either greedy-navigable or constant neighborhood relations. However, for some graph classes, these relations are relatively easy to compute. For example, for a planar graph, the intersection one-hop closed neighborhood relation the neighborhood for each node v only need to contain its adjacent nodes. This relation is also constant and navigable. In quasi-unit-disk graphs where the inner radius r is greater than $1/\sqrt{2}$, Barrière et al [3] proved the following: for every edge (u, v) and any edge (w, x) intersecting (u, v) either w or x are adjacent to either u or v. That is, a two-hop neighborhood for each node forms a constant two-hop intersection closed neighborhood relation.

Translation from planar to non-planar routing algorithm. We focus on constructing a non-planar routing algorithm NA on the basis of an existing planar routing algorithm PA. The construction is as follows. A planar graph PG is created by adding a *virtual node* at every edge intersection of the non-planar graph NG. The algorithm NA simulates the execution of the planar routing algorithm PA on this planar graph PG.

In the non-planar algorithm NA, the steps of the translated PA are executed in *real nodes*: the original nodes of the non-planar graph NG. Specifically, the NA continuously repeats these two stages: determine which node of PG to send the message next according to PA; and, (II) route the message to the node in NA that will evaluate the next step of PA.

The algorithm translation may potentially be rather inefficient. For example, the message may have to visit several nodes in NG to collect sufficient information for the Stage I of the translation: the decision as to which node in PG, the message needs to be sent. Similarly, the message may potentially visit multiple nodes of NG to gather routing information for the navigation of Stage II.

We are interested in constructing efficient algorithm translations which we call direct. A translation is *direct* if Stage I of PA is evaluated at a single node of NG; and in Stage II, on its way from one node of PG to the next, the message visits each node of NG at most once. In effect, a direct-translation algorithm provides a mapping from nodes of PG onto nodes of NG for planar algorithm routing decisions and then routes the message from one such mapped non-planar node to the next.

Theorem 1 *If an arbitrary planar geometric routing algorithm* PA *guarantees delivery in any planar graph, then there exists its direct translation to a non-planar geometric routing algorithm* NA *that also guarantees delivery for any non-planar graph NG, provided that the neighborhood relation $\mathcal{N}(NG)$ is two-hop closed and either constant or greedy navigable.*

Proof: We prove the theorem by constructing the non-planar routing algorithm NA using direct translation.

Recall that the nodes of PG are formed by edge intersections of NG. In the constructed algorithm, a node u of NG executes Stage I for every node p of PG that is formed by intersecting an edge incident to u (see Figure 2).

By definition, if the neighborhood relation $\mathcal{N}(NG)$ is two-hop closed, the neighborhood $N(u)$ of node u contains every edge (u, v), incident to u, every edge (w, x) that intersects (u, v) as well as every edge (y, z) that

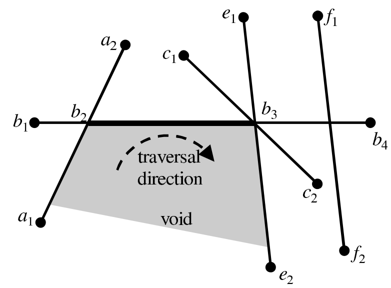

Figure 3: Void traversal example. Real head node b_1 receives the message traversing the void from previous head node a_1; determines the virtual node b_3 that ends its segment bordering the void and begins the next segment; and forwards the message to the next real head node e_1.

intersects (w, x). That is, the neighborhood of u contains all edges incident to node p in the planar graph PG.

This means that node u has sufficient information to execute Stage I of the translation, i.e. to execute PA for node p and decide which node q of PG to route the message next. In other words, which node r of NG shall evaluate Stage I for q.

Let us consider the routing of the message from node u to r, i.e. the Stage II of the translation algorithm. By the definition of two-hop closure of $\mathcal{N}(NG)$, neighborhood $N(u)$ contains the route to r. According to the conditions of the theorem, $\mathcal{N}(NG)$ is either constant or navigable. Due to Proposition 1, in either case, it is possible to route the message from u to r without the message exceeding a limit that is independent of the graph G size. That is, the Stage II of the constructed $\mathcal{NG}$ satisfies message size requirements of geometric routing.

To summarize, if the original planar geometric routing algorithm PA guarantees delivery, so does the constructed non-planar geometric routing algorithm NG. $\square$

Theorem 1 guarantees the existence of a direct translation regardless of the planar routing algorithm. This translation requires that the neighborhood relation be two-hop closed. The computing infrastructure may not be able to accommodate memory requirements of such neighborhood relation.

For specific algorithms, these memory requirements may be reduced. For this reduction, we have to place limits on the direct translation. The translation of a geometric routing is *greedy direct* if: switching between greedy and face routing mode occurs only on real nodes and greedy navigation proceeds only through real nodes.

Proposition 2 *Greedy-direct translation does not violate correctness of* FACE*,* GFG*,* OAFR *or* GOAFR+*.*

Theorem 2 *There exists a greedy-direct translation of* FACE*,* GFG*,* OAFR *and* GOAFR+ *to non-planar routing algorithms, provided that the neighborhood relation is one-hop closed and either constant or greedy navigable.*

Proof: We prove the theorem for *GFG* as the most illustrative case. The proof for the other algorithms is similar. As in the proof of Theorem 1, we need to provide a mapping from every virtual node to a real node such that the real node has sufficient information for Stage I of the translation (making the decision as to what the next node should be in the planar routing algorithm) and then execute Stage II (forwarding the message to the mapping of this next node).

This translation depends on the particular decisions that *GFG* makes during the execution. For *GFG*, the decisions are: switching from greedy to face mode and back, greedy routing, face traversal and switching between faces.

Since the theorem considers greedy-direct translation, switching between modes and greedy routing is done at real nodes only. Hence, the translation from planar to non-planar algorithm is trivial as the node of the planar graph that the planar geometric routing algorithm navigates are immediately mapped to the real nodes of the non-planar graph.

We, therefore, need to provide the translation for face traversal and face switching. The process of face traversal in the translated algorithm is, in effect, the void traversal over virtual nodes. Each virtual node is a virtual head node of a segment bordering this void. We map this node to the real head node of this segment.

Let us first consider how this translation operates for face traversal. Since the conditions of this theorem specify that the neighboring relation $\mathcal{N}(NG)$ is one-hop closed, the real head node of the segment contains the information about all the edges intersecting the edges incident to the real head node. That is, this real head node is capable of determining the virtual tail node of this segment, which is the virtual head node of the next segment. Hence, the real head node is able to detect the real head node of the next segment which is also present in its neighborhood. In other words, this real head node is capable of executing Stage I of the translation.

Since the neighborhood relation is either constant or greedy navigable, due to Proposition 1, it is impossible to route the message from from the real head node to the next without the message exceeding a limit that is independent of the graph G size. That is, the Stage II of the translation satisfies message size requirements of geometric routing.

Let us now consider switching faces. Algorithm *GFG* uses after-crossing face switching. That is, when the edge in the planar graph crosses the sd-line, to start the traversal of the next face, the previous and next nodes incident to this edge exchange places, while traversal direction (left or right-hand) remains the same.

To execute Stage I of the translation, the real head node needs to determine if the segment crosses the sd-line. For that, just like in case of face routing, it is necessary for this real head node to determine the end of the segment; and, just like in case of face routing, one-hop closed neighborhood relation $\mathcal{N}(NG)$ provides sufficient information for that.

Once Stage I is executed and the face switching is decided, Stage II – the actual face switching needs to take place. However, this amounts to the real head node just forwarding the message to the other node incident to the edge containing the traversed segment. That is, Stage II of the transformation in this case also satisfies message requirements of the geometric routing.

So summarize, for every mode of *GFG*, provided it is a greedy-direct translation, we supplied a mapping from virtual to real nodes that, in case the neighborhood relation is one-hop closed and either constant or greedy navigable, enables the execution of Stage I and Stage II of the translated algorithm. Hence the theorem. □

Void traversal illustration. Let us illustrate void traversal used in the proof of Theorem 2 (refer to Figure 3) The head real node b_1 receives a message that traverses the void using left-hand-rule from the previous head node a_1. This indicates b_2 the intersection of the edges $(b1, b4)$ and (a_1, a_2) is the virtual head node b_2 of the segment bordering to the traversed void. Node b_1 needs to determine the end of this segment. For that, b_1 searches for the edge that intersects (b_1, b_4) closest to to b_2 in the direction of traversal. If several edges intersect (b_1, b_4) in the same place, b_1 selects the edge that makes the smallest angle with (b_1, b_4) in the counter-clockwise direction. This edge happens to be (e_1, e_2) that intersects (b_1, b_4) at point b_3 forming a virtual tail node that completes segment (b_3, b_4). Node b_3 is also a virtual head node for the next segment. The real head node for the next segment is e_1. Hence, b_1 forwards the message to e_1 for further pressing.

Figure 4, illustrates the operation of the complete translation of *FACE* algorithm. We call this translation *VOID*. The communication graph in this figure is a non-planar version of the graph shown in Figure 1. Since the graph is not planar, some of the edge intersection points such as f, g, and k are virtual nodes. The source starts a clockwise traversal of void V_1 by navigating segment (s, f). Node s is both a virtual and a real head

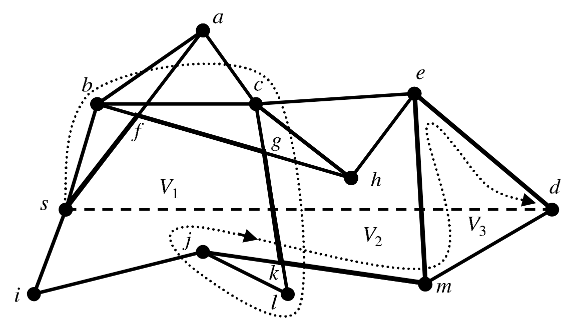

Figure 4: *VOID* example routing. The dotted line is the path taken by the message. Bold line highlight the traversed edge segments adjacent to the voids.

node of this segment. Node s determines the tail of this segment and the head of the next segment to be virtual node f. Node s then forwards the message to the real head node of the next segment to node b. Node b forwards the message to c which is a head node of the next segment. Node c determines that the next segment intersects sd-line, switches voids and forwards the message to j which is the real head node of the next segment. Note that c is not adjacent to j, so c has to forward the message through an intermediate node l. In this manner, the message is routed to m, e and then d.

4 Performance Evaluation

Simulation environment. To validate the effectiveness of the concept of void routing, we recreated the abstract simulation environment of Kuhn et al [21, 24] and instrumented it to utilize non-unit disk graphs. Specifically, we used quasi-unit-disk adjacency model with probability $p = 0.5$ and inner radius $r = 0.75$. For our experiments we used 20×20 units simulated square field randomly filled by the nodes with randomly selected source and destination pairs. We used 21 different density levels from from 0.3 to 20 nodes per unit disk. The number of nodes in each generated graph depended on the selected density level. For each graph, a single source and destination pair was randomly selected.

To validate our environment we measured the same graph parameters as Kuhn et al [21, Figure 3], [24, Figure 3]: shortest path span (ratio between Euclidean and path distance), the ratio of connected graphs, and the rate of success of greedy routing. For each density level we carried out $2,000$ measurements. The obtained results are plotted in Figure 5. The shape of the graphs is similar to those obtained by Kuhn et al. Recall that

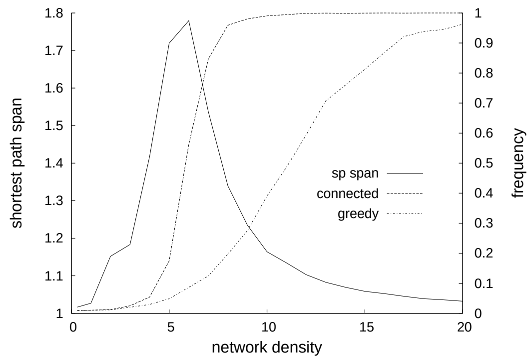

Figure 5: Graph parameters depending on its density. Shortest path span, ratio of connected graphs, and rate of success of pure greedy routing. Last two plotted against the right y axis.

Kuhn et al defined *critical density* range as the range where the length of the shortest path between the source and destination significantly deviates from the Euclidean distance between them. The critical density range is the main area of routing optimization. Note that since some of the nodes within the quasi-unit-disk are only probabilistically connected, the quasi-unit-disk based graphs are sparser than plain unit-disk based ones. Thus, the critical density of our graphs is somewhat shifted to the right.

We evaluated the performance of geometric routing algorithms on “raw” non-planar graphs that we generated and on Gabriel subgraphs of these graphs.

Evaluation description. We translated the planar routing algorithms *FACE*, *OAFR*, *GFG* and *GOAFR+* to the non-planar versions which we call *VOID*, *VOAFR*, *GVG* and *VGOAFR+* respectively.

For each algorithm and for each graph density level, we performed 2,000 measurements on raw graphs. The performance achieved by these algorithms is plotted in Figure 6. For performance evaluation, we use the same metric as Kuhn et al [21, 22, 24]: the ratio of the path between the source and destination selected by the algorithm and the shortest path. That is, *mean performance* in the plotted graphs is the average such ratio.

To study the effect of graph scale on the performance of geometric routing algorithms, we constructed the simulation scenario similar to that of Kuhn et al [24, Figure 10]. We fixed the density of the graph near the critical value — at 6 and varied the field size. Specifically, we selected 10 different lengths of the side of the square field from 4 to 40 units. The number of nodes in the field was selected to match the required density of

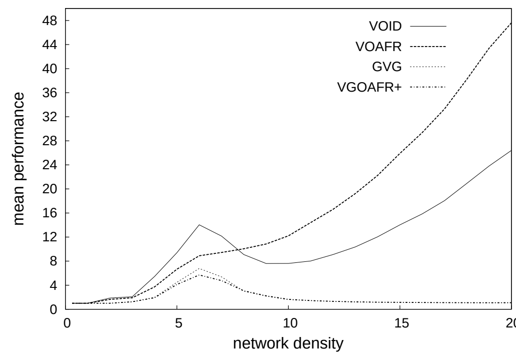

Figure 6: Mean performance of geometric routing algorithms on raw quasi-unit-disk graphs graphs depending on communication graph density.

6. We took 3,000 measurements for each side length. The simulation results are shown in Figure 7.

Recall that on planar graphs *OAFR/GOAFR+* is known to significantly outperform other geometric routing algorithms such as *FACE/GFG*. Interestingly, on non-planar graphs, its advantage is far less significant. The reason is that *OAFR/GOAFR+* uses a number of parameters, such as the size of the ellipse around the source-destination pair, that optimize its performance for planar graphs. We, however, used the parameters stated in the original papers that presented the algorithms. Potentially, these parameters could be tuned to optimize the performance of *OAFR/GOAFR+* for non-planar graphs. Note also that as the density of the graph increases, the performance of standalone face traversal algorithms *VOID* and *VOAFR* deteriorates. This, however, is not a concern as the algorithms will be used in combination with greedy routing in practice. With the density increase, the performance of combined algorithms *GVG* and *VGOAFR* approaches optimal.

We repeated the experiments on Gabriel subgraphs of the simulated graphs. Recall that to ensure connectivity, the mutual witness has to be used in creating these subgraphs. The results of the density and scale experiments are shown in Figures 8 and 9 respectively. Gabriel subgraphs tend to be significantly sparser than the raw graphs. This lack of edges adversely affects the performance of void routing algorithms. However, this performance penalty may be justified since, due to fewer edges, each node may have lower computation and storage load to make routing decisions.

Our experiments indicate that the translation that we describe in this paper is readily applicable to geometric routing algorithms. The resultant algorithms performed successfully over a variety of simulated communication

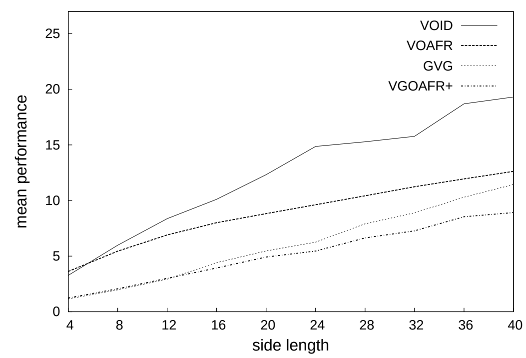

Figure 7: Mean performance of geometric routing algorithms on raw quasi-unit-disk graphs depending on graph scale.

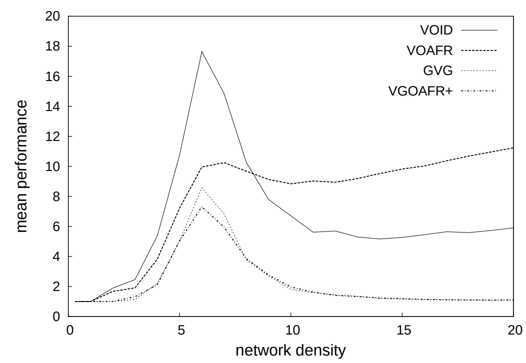

Figure 8: Mean performance of geometric routing algorithms on Gabriel subgraphs depending on communication graph density.

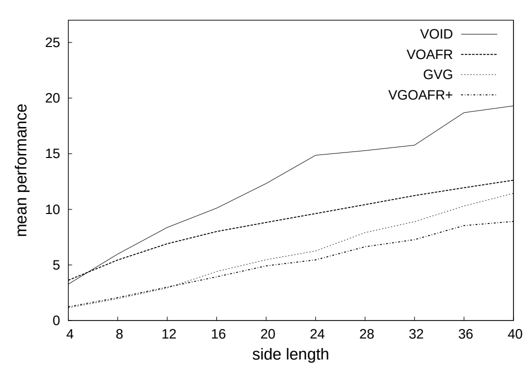

Figure 9: Mean performance of geometric routing algorithms on Gabriel subgraphs depending on graph scale.

graphs. Our results display the performance dynamics of geometric routing algorithms which can allow the engineers to select proper network parameters for the behavior of the algorithms on their systems.

Mobility. To evaluate the performance of our algorithm in non-stationary network, we added mobility to the simulation environment of Kuhn et al. Specifically, we implemented random waypoint mobility model [15]. Each node selects a waypoint in the simulated square field uniformly at random. The node then uniformly at random selects its speed from zero to a specified maximum velocity and proceeds to the selected waypoint where it pauses for random amount of time before selecting the next waypoint. We kept the rest of the parameters of simulation model of Kuhn et al. as described earlier. In particular, we used the quasi-unit-disk radio model.

The speed of the node is expressed in units of distance per unit of time. The message transmission was computed as follows. A single message transmission is assumed to take one unit of time. After message receipt, the node positions are computed according to the distance the nodes traveled during the unit of time it took to transmit the message; the routing decision is made according to the routing algorithm, the message is then transmitted to the next node and the process repeats. We measured the *success rate* of a routing algorithm. We assumed that the algorithm succeeds in delivering the message to the destination node if the message arrives at the initial destination location before this node moves more than one unit distance away from this location.

For our evaluation, we selected the network at critical density of 4.71 (approximately 1.5π). We selected 26 maximum velocities of the random waypoint mobility model and carried out $1,000$ measurement for each velocity. As with the static case, the source and destination pairs were randomly selected.

We conducted the evaluation for *VOID* and *GVG*. The results of our evaluation are shown in Figure 10. The results indicate that *GVG* has greater success rate than *VOID* due to the shorter routes selected by *GVG* during its greedy routing phase. As the speed of the nodes increases, the probability of successful message delivery diminishes.

5 Conclusion

Scalability makes geometric routing an attractive choice for navigation in ad hoc wireless networks. Traditional geometric routing requires a planar subgraph for local minima recovery. Planarizing a communication graph induced by a realistic wireless network communication may be so expensive as to nullify the advantages of

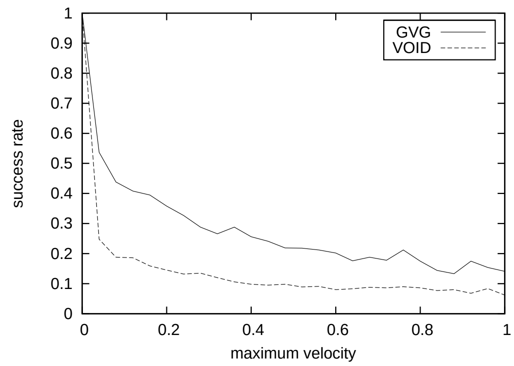

Figure 10: Success rate of geometric routing algorithms under random waypoint mobility model depending on maximum node speed.

geometric routing itself.

We stated sufficient conditions for translating planar routing algorithms to non-planar routing algorithms. We then translated several popular planar routing algorithm to non-planar versions and evaluated their performance.

In closing we would like to point out a few possible research directions resulting from our work. Theorems 1 and 2 state sufficient memory conditions for the existence of planar to non-planar algorithm translation. It, therefore, is interesting study the minimum storage bounds that still allow geometric routing on non-planar graphs. Another interesting research direction is to compare the performance of *VOID*-based geometric routing algorithms to the algorithms based on direct extensions of planarization techniques proposed elsewhere [17, 19, 23]. To further increase the practicality of geometric routing, it is worth investigating the stability of *VOID*-based routing algorithms to radio propagation irregularities such as asymmetric links or errors in topology information [11, 35].

Acknowledgments

We would like to thank Ivan Stojmenovic of the University of Ottawa for his helpful comments.

References

[1] I. Abraham, D. Dolev, and D. Malkhi. LLS: a locality aware location service for mobile ad hoc networks. In *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 75–84, Philadelphia, USA, October 2004. ACM.

[2] Anish Arora, Rajiv Ramnath, Emre Ertin, Prasun Sinha, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Hui Cao, Mukundan Sridharan, Santosh Kumar, Nick Seddon, Chris Anderson, Ted Herman, Nishank Trivedi, Chen Zhang, Mikhail Nesterenko, Romil Shah, Sandeep S. Kulkarni, Mahesh Aramugam, Limin Wang, Mohamed G. Gouda, Young ri Choi, David E. Culler, Prabal Dutta, Cory Sharp, Gilman Tolle, Mike Grimmer, Bill Ferriera, and Ken Parker. Exscal: Elements of an extreme scale wireless sensor network. In *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 102–108, August 2005.

[3] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny. Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. *Wireless Communications and Mobile Computing*, 3(2):141–153, 2003.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *The Journal of Mobile Communication, Computation and Information*, 7(6):48–55, 2001.

[5] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. Self-configuring localization systems: Design and experimentałevaluation. *ACM Transactions on Embedded Computing Systems*, 3(1):24–60, February 2004.

[6] Thomas Clouser, Mark Miyashita, and Mikhail Nesterenko. Fast geometric routing with concurrent face traversal. In *Principles of Distributed Systems, 12th International Conference, (OPODIS)*, pages 346–362, December 2008.

[7] S. Datta, I. Stojmenovic, and J. Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. *Cluster Computing*, 5(2):169–178, April 2002.

[8] G.G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, March 1987.

[9] H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, MOBICOM 2006, Los Angeles, CA, USA, September 23-29, 2006*, pages 390–401. ACM, 2006.

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

[11] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report CSD-TR 02-0013, UCLA, 2002.

[12] S. Giordano, I. Stojmenovic, and L. Blazevic. Position based routing algorithms for ad hoc networks - a taxonomy. *Ad Hoc Wireless NetWorking*, pages 103–136, January 2004.

[13] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.

[14] J.L. Hill and D.E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, November/December 2002.

[15] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353, pages 153–181. 1996.

[16] B. Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobille Computing and Networking (MobiCom 2000)*, pages 243–254. ACM Press, August 2000.

[17] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *2nd Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, USA, May 2005.

[18] Y-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *3d ACM/SIGMOBILE International Workshop on Foundations of Mobile Computing (DIAL-M-POMC)*, pages 34–43, 2005.

[19] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Lazy cross-link removal for geographic routing. In *Proceedings of the 4th international conference on Embedded networked sensor systems SenSys*, pages 112–124, Boulder, Colorado, USA, 2006.

[20] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. 11 th Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.

[21] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. *22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, July 2003.

[22] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, Atlanta, Georgia, USA, September 2002.

[23] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *Joint Workshop on Foundations of Mobile Computing (DialM-POMC)*, pages 69–78, San Diego, CA, USA, September 2003.

[24] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *4th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 267–278, Annapolis, Maryland, USA, June 2003.

[25] B. Leong, B. Liskov, and R. Morris. Geographic routing without planarization. In *3rd Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, USA, May 2006.

[26] J. Li, J. Jannotti, D.S.J. De Couto, D.K. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 120–130, August 2000.

[27] Wen-Jiunn Liu and Kai-Ten Feng. Greedy routing with anti-void traversal for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 8(7):910–922, 2009.

[28] Joseph P. Macker and M. Scott Corson. Mobile ad hoc networking and the IETF. *Mobile Computing and Communications Review*, 6(2):1–2, 2002.

[29] An Nguyen, Nikola Milosavljevic, Qing Fang, Jie Gao, and Leonidas J. Guibas. Landmark selection and greedy landmark-descent routing for sensor networks. In *26th Annual IEEE Conference on Computer Communications (INFOCOM)*, pages 661–669, May 2007.

[30] Ananth Rao, Christos H. Papadimitriou, Scott Shenker, and Ion Stoica. Geographic routing without location information. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, (MOBICOM)*, pages 96–108, September 2003.

[31] Stefan Ruhrup and Ivan Stojmenovic. Contention-based georouting with guaranteed delivery, minimal communication overhead, and shorter path in wireless networks. In *International Symposium on Parallel and Distributed Processing (IPDPS)*, pages 1–9, April 2010.

[32] K. Seada, A. Helmy, and R. Govindan. On the effect of localization errors on geographic face routing in sensor networks. In *Proceedings of the third international symposium on Information processing in sensor networks (IPSN)*, pages 71–80, New York, April 26–27 2004. ACM Press.

[33] G.T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[34] A. Vora and M. Nesterenko. Void traversal for guranteed delivery in geometric routing. *The 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2005)*, pages 63–67, November 2005.

[35] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks*, 2(2):221–262, 2006.